1980

A two-dimensional unsteady Euler equation solver for flows in arbitrarily shaped regions using a modular concept

Richard Glenn Hindman
Iowa State University

HINDMAN, RICHARD GLENN

A TWO-DIMENSIONAL UNSTEADY EULER EQUATION SOLVER FOR FLOWS IN ARBITRARILY SHAPED REGIONS USING A MODULAR CONCEPT

Iowa State University PH.D. 1980

A two-dimensional unsteady Euler equation solver for flows in arbitrarily shaped regions using a modular concept

by

Richard Glenn Hindman

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1980

TABLE OF CONTENTS

CHAPTER I. INTRODUCTION. 1
Modular Approach 2

CHAPTER II. GOVERNING EQUATIONS. 8

CHAPTER III. GEOMETRY. 11
Grid and Gridspeed Operators 14

CHAPTER IV. COUPLING OF GEOMETRY TREATMENT AND FINITE-DIFFERENCE SCHEME 18

CHAPTER V. ALGORITHM 22

CHAPTER VI. INITIAL AND BOUNDARY CONDITIONS. 24
Initial Conditions 24
Boundary Conditions. 24
Body boundary conditions. 25
Shock boundary conditions 25
Slip surface boundary conditions. 27
Other boundary conditions 31
Corner Conditions. 31
Simple intersection 32
Complex intersection. 34
General Corner Speed 36
General Boundary Point Speed 38

CHAPTER VII. NUMERICAL RESULTS 41
A Single Module. 41
Two Modules. 52

CHAPTER VIII. CONCLUDING REMARKS 59

ACKNOWLEDGMENTS 60

REFERENCES. 61

APPENDIX: ANALYSIS OF THE GRID PROPAGATION TECHNIQUE AND ITS COUPLING TO THE GOVERNING EQUATIONS. 65

CHAPTER I. INTRODUCTION

The Electronic Revolution which undoubtedly is still in its infancy has given man an incredibly powerful tool with which to solve many of the problems faced in today's highly technological society. This document deals with the application of this tool, the high speed digital computer, to problem solving in fluid dynamics.

Since the advent of the high speed digital computer, an extensive effort has been made toward obtaining solutions to fluid dynamics problems for which there exist concise mathematical descriptions yielding a system of equations and boundary conditions that represent an approximation to various physical processes. Many different mathematical descriptions exist depending upon the nature of the assumptions inherent in the derivation. The present work concentrates on fluid dynamics problems in which the flow is assumed to have a compressible and nonviscous nature. Such flow problems are adequately described by the Euler equations. An additional simplification (not restriction) which is made in the present work is that of two-dimensional flow.

There are many computer codes in existence for solving the two-dimensional Euler equations (1-3). Unfortunately, each of them was written with a particular class of problems in mind. This places restrictions on the range of applicability of any computer code. Such restrictions may be categorized as those pertaining to the physics of the flow and those pertaining to the problem geometry description. These two categories are not necessarily independent. The former category in-

cludes such things as the development of flow singularities, discontinuities, or steep gradients for which there is inadequate numerical treatment. The latter category involves the manner in which the flow field grid point distribution is defined relative to a base coordinate system. For example, a typical restriction of this type arises when a shock boundary is defined in Cartesian variables as $x_s = x_s(y,t)$ and possibly due to some interaction process, the shock slope, $\partial x_s/\partial y$, becomes unbounded at some point. This difficulty can certainly be remedied in a given situation by a coordinate rotation or some other trick but in the general case any such trick, which may well require extensive code modification, will undoubtedly have similar limitations of its own. It is the purpose of this research effort to lift some of these restrictions by developing a generalized two-dimensional Euler equation solver using a modular approach and a very general treatment of the module geometry thus providing one computer code capable of solving a wide variety of two-dimensional fluid dynamics problems.

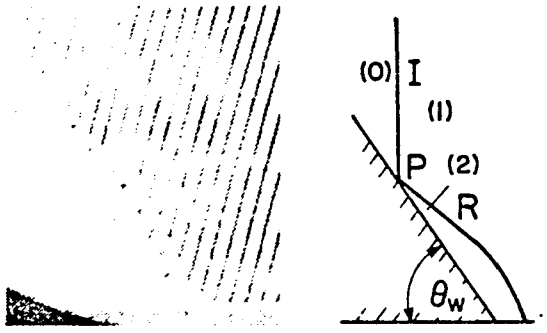
Modular Approach

Although the present approach and the resulting computer code are in no way limited to supersonic flow problems, such problems do provide a more extensive test of the general concepts developed. As a result, the discussion presented herein will tend to emphasize the application of the present approach to supersonic flow.

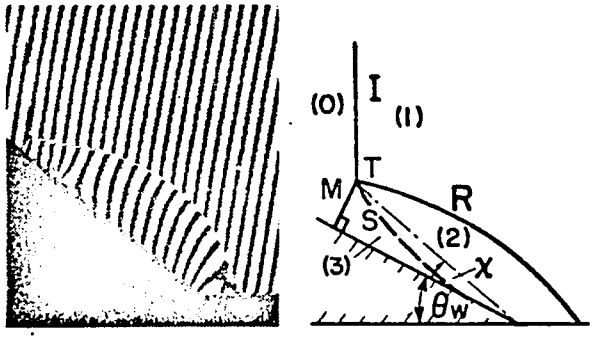
Many steady and unsteady supersonic flow problems contain multiple

regions of continuous flow, each of which is either bounded by a surface across which the flow is discontinuous such as a shock wave or a slip surface or bounded by an impermeable surface such as the body. For example, consider the classical planar shock diffraction problem depicted schematically in Figure 1 (4). Four different wave patterns are shown which illustrate the existence of various distinct flow regions or modules. Such behavior is typical of virtually all inviscid shock interaction problems. The present effort is to develop a computational solver which is designed to compute the solution to the Euler equations in an arbitrary module. The complete flow problem may consist of many of these modules coupled together through the appropriate application of boundary condition procedures. For example, the single Mach reflection problem may be described with two modules as shown in Figure 2. These modules share a slip surface as a common boundary.

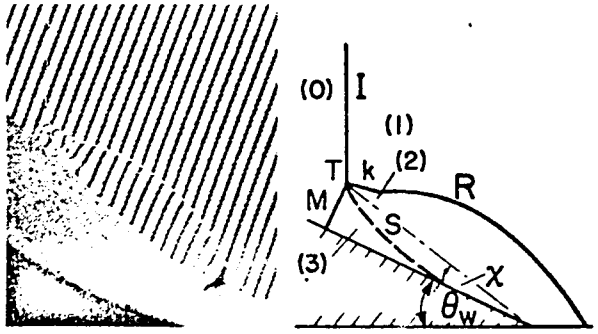
Since the present approach requires that the flow module boundaries also be computational boundaries a generalized mapping of the independent variables must be performed (see Figure 3). It is clear that in the general case these module boundaries are time-varying in nature. Such boundaries must therefore be capable of assuming virtually any shape dictated by the governing equations. Consequently, it is especially important that there is no built in dependence of the validity of the module geometry description (grid point distribution and movement) on the particular base coordinate system chosen as a reference frame. The unique manner in which the present technique avoids such



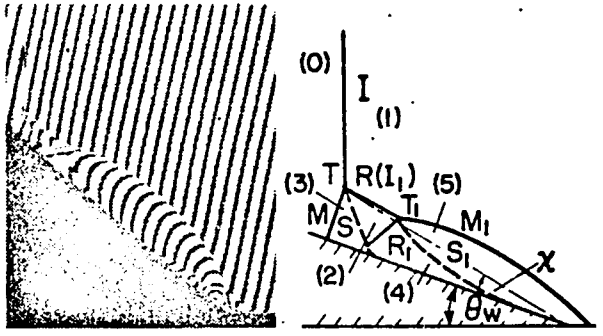
a. Regular Reflection (RR), Wedge Angle θ_W = 60 deg, Shock Mach Number M_S = 4.71



b. Single-Mach Reflection (SMR), θ_W = 40 deg, M_S = 2.10



c. Complex-Mach Reflection (CMR), θ_W = 40 deg, M_S = 2.59



d. Double-Mach Reflection (DMR), θ_W = 40 deg, M_S = 4.78

Figure 1. Four possible shock-diffraction wave patterns

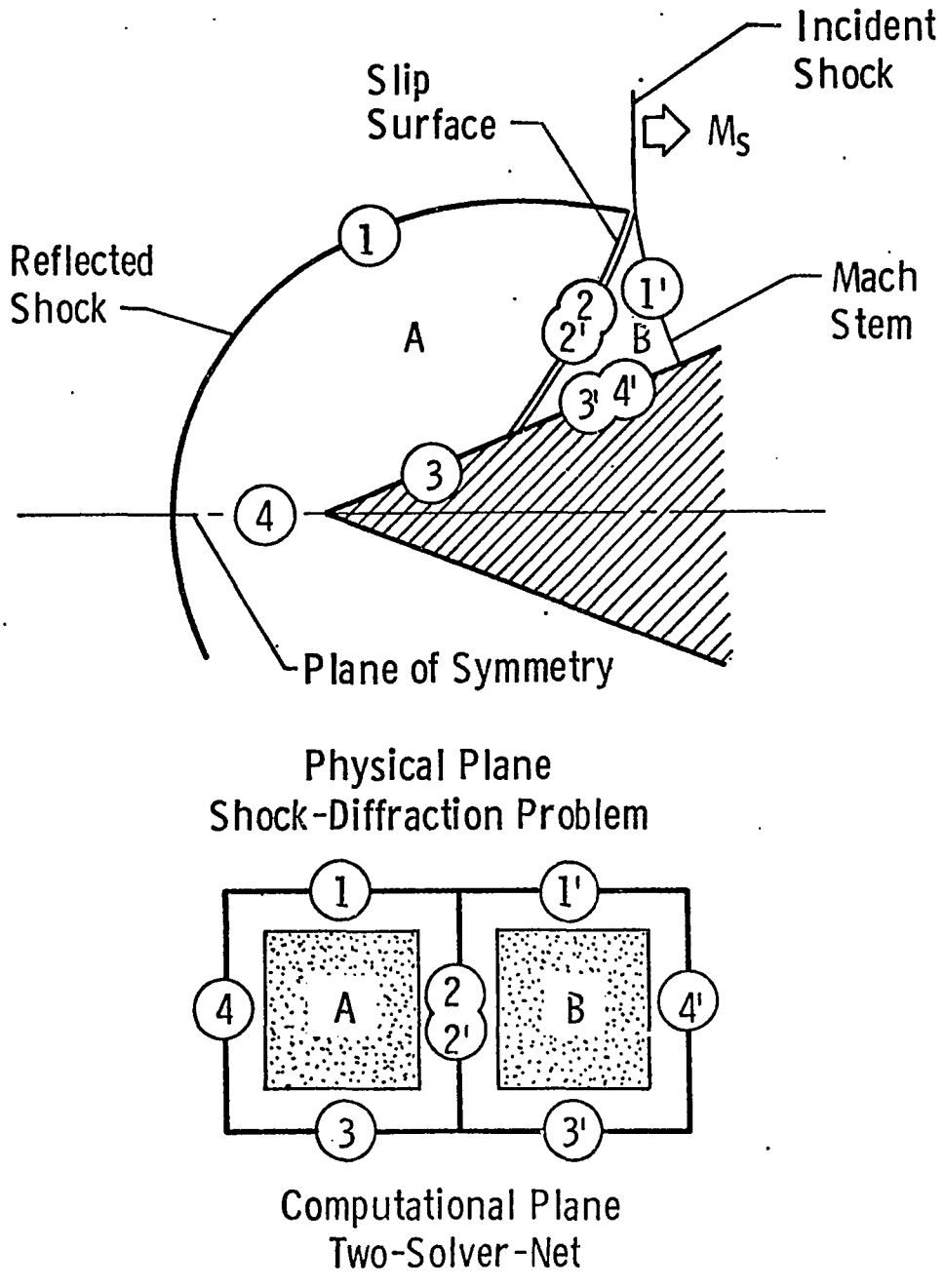


Figure 2. Conceptual network of solvers

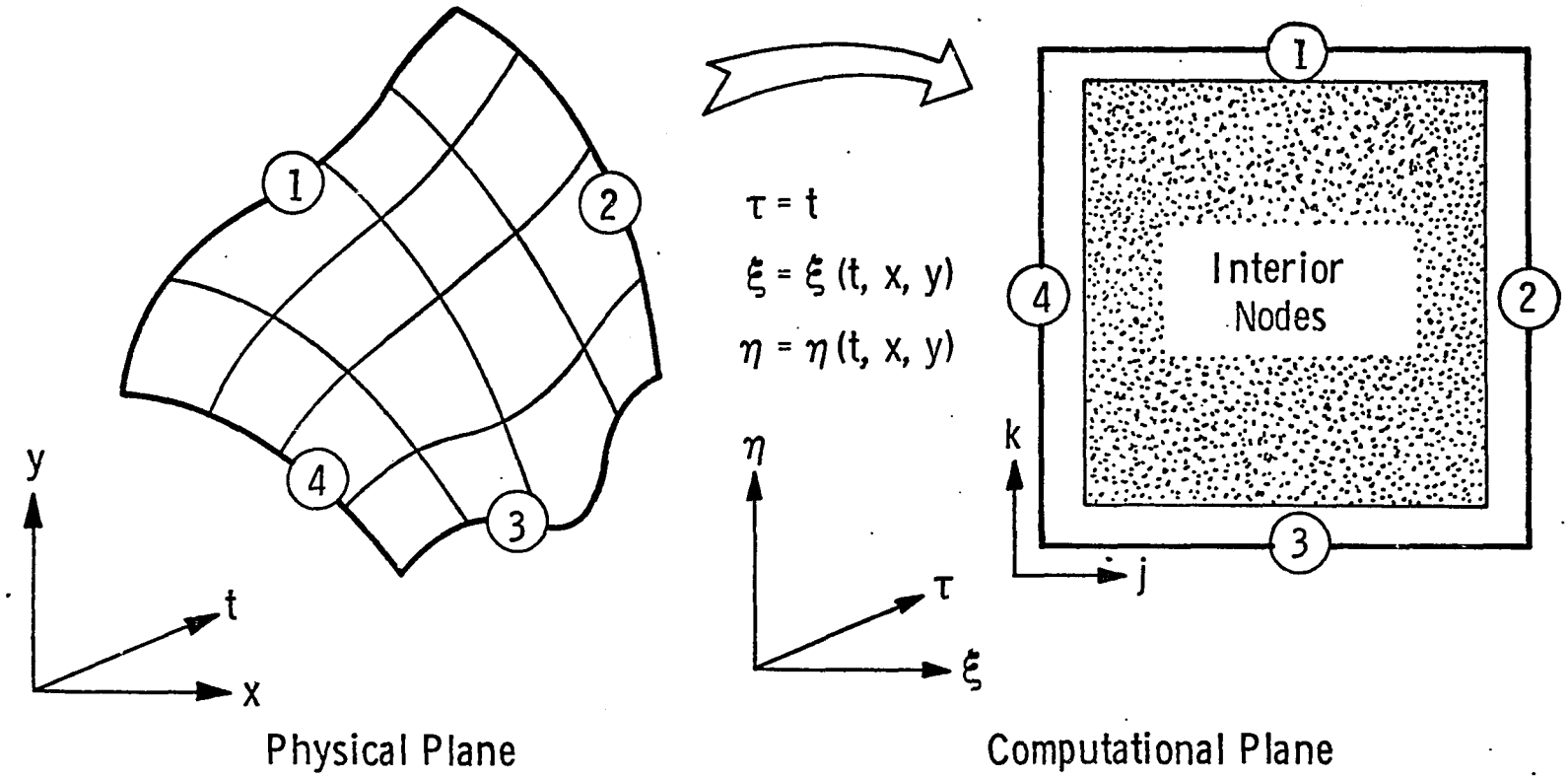


Figure 3. Transformation of physical plane to computational plane for single solver

dependence is described in a later section.

A literature search reveals that while extensive information exists on the idea of patching together of solutions, for example in boundary layer-inviscid interactions, etc., very little information is available on the present modular type of approach. Ludloff and Friedman (5) solve the Mach reflection planar shock diffraction problem with what is apparently two modules although they do not specifically indicate such an approach.

In contrast to the lack of information available on the modular type approach, considerable information is available on the subject of generalized geometry. Various types of automatic grid generation procedures have been developed (6-15) some of which have been applied to domains with moving boundaries (1,13,16). The present approach (17) uses the automatic grid generation procedure of Thompson, Thames, and Mastin (6) and extends it in a unique way to allow for domains with time varying as well as arbitrarily shaped boundaries.

CHAPTER II. GOVERNING EQUATIONS

The two-dimensional unsteady Euler equations are written in conservation-law form in Cartesian coordinates as

$$\frac{\partial \bar{q}}{\partial t} + \frac{\partial \bar{f}}{\partial x} + \frac{\partial \bar{g}}{\partial y} = 0 \qquad 1$$

where $\bar{q}$, $\bar{f}$, and $\bar{g}$ are given by

$$\bar{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad \bar{f} = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ (p + e)u \end{pmatrix}, \quad \bar{g} = \begin{pmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ (p + e)v \end{pmatrix}$$

with (u,v) representing the Cartesian (x,y) velocity components, ρ the density, p the static pressure, and e the total energy per unit volume. e is related to p, ρ, u, v through the equation

$$e = \frac{p}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2)$$

where γ is the ratio of thermal capacities of the fluid.

Cartesian coordinates are used as the base coordinate system but in order to map bodies and other surfaces on to constant coordinate lines, the following coordinate mapping is introduced (see Figure 3).

$$\begin{aligned} \tau &= t \\ \xi &= \xi(t,x,y) \\ \eta &= \eta(t,x,y) \end{aligned} \qquad 2$$

The governing equations are transformed by this mapping according to Viviand (18) into a new strong conservation-law form written as

$$\frac{\partial \overline{Q}}{\partial \tau} + \frac{\partial \overline{F}}{\partial \xi} + \frac{\partial \overline{G}}{\partial \eta} = 0 \qquad 3$$

where

$$\begin{aligned} \overline{Q} &= I\overline{q} \\ \overline{F} &= I(\xi_t \overline{q} + \xi_x \overline{f} + \xi_y \overline{g}) \\ \overline{G} &= I(\eta_t \overline{q} + \eta_x \overline{f} + \eta_y \overline{g}) \end{aligned} \qquad 4$$

and

$$\begin{aligned} I &= x_\xi y_\eta - x_\eta y_\xi = \frac{1}{J} \\ J &= \xi_x \eta_y - \xi_y \eta_x \end{aligned}$$

J is the Jacobian of the mapping and I is the Jacobian of the inverse mapping. The metrics of the mapping are related to the metrics of the inverse mapping by the equations

$$\begin{aligned} I\xi_t &= y_\tau x_\eta - x_\tau y_\eta \\ I\xi_x &= y_\eta \\ I\xi_y &= -x_\eta \\ I\eta_t &= x_\tau y_\xi - y_\tau x_\xi \\ I\eta_x &= -y_\xi \\ I\eta_y &= x_\xi \end{aligned} \qquad 5$$

Using Equation 5, Equation 4 is rewritten as

$$\begin{aligned} \overline{Q} &= I\overline{q} \\ \overline{F} &= (y_\tau x_\eta - x_\tau y_\eta)\overline{q} + y_\eta \overline{f} - x_\eta \overline{g} \\ \overline{G} &= (x_\tau y_\xi - y_\tau x_\xi)\overline{q} - y_\xi \overline{f} + x_\xi \overline{g} \end{aligned} \qquad 6$$

Conservation-law form of the governing equations is necessary according to Lax (19) to ensure that the jump conditions existing across weak solutions are automatically satisfied. This form of the governing equations thus adds a shock-capturing capability to the resulting computer code.

The governing equations are transformed by this mapping according to Viviand (18) into a new strong conservation-law form written as

$$\frac{\partial \bar{Q}}{\partial \tau} + \frac{\partial \bar{F}}{\partial \xi} + \frac{\partial \bar{G}}{\partial \eta} = 0 \qquad 3$$

where

$$\begin{aligned} \bar{Q} &= I\bar{q} \\ \bar{F} &= I(\xi_t \bar{q} + \xi_x \bar{f} + \xi_y \bar{g}) \\ \bar{G} &= I(\eta_t \bar{q} + \eta_x \bar{f} + \eta_y \bar{g}) \end{aligned} \qquad 4$$

and

$$\begin{aligned} I &= x_\xi y_\eta - x_\eta y_\xi = \frac{1}{J} \\ J &= \xi_x \eta_y - \xi_y \eta_x \end{aligned}$$

J is the Jacobian of the mapping and I is the Jacobian of the inverse mapping. The metrics of the mapping are related to the metrics of the inverse mapping by the equations

$$\begin{aligned} I\xi_t &= y_\tau x_\eta - x_\tau y_\eta \\ I\xi_x &= y_\eta \\ I\xi_y &= -x_\eta \\ I\eta_t &= x_\tau y_\xi - y_\tau x_\xi \\ I\eta_x &= -y_\xi \\ I\eta_y &= x_\xi \end{aligned} \qquad 5$$

CHAPTER III.. GEOMETRY

There appears to be no universal procedure in the literature for treating the metric and Jacobian terms appearing in Equation 6. This section, therefore, attempts to identify some ground rules which are followed in the treatment of these terms. The concepts involved are more completely discussed by Steger (1). The rules stem from accuracy considerations and are based on the intuitive suggestion that for a scheme to be considered a good one, the flowfield code which it supports must be capable of exact reproduction of a uniform flow. That is, with the boundary values held fixed at some uniform flow conditions and the initial flow field set to these same conditions, the finite-difference algorithm should exactly reproduce this same flow field for all time. This is actually a statement of the independence that should exist between the flow and the grid distribution geometry for the case of uniform flow. It provides a simple test for an existing numerical algorithm and provides a criterion which ties down many of the questions arising in code development work regarding the manner in which the metrics of Equation 6 should be computed.

Consider the expanded form of Equation 3,

$$\frac{\partial}{\partial\tau}\left[I\bar{q}\right] + \frac{\partial}{\partial\xi}\left[(y_\tau x_\eta - x_\tau y_\eta)\bar{q} + y_\eta \bar{f} - x_\eta \bar{g}\right]$$

$$\frac{\partial}{\partial\eta}\left[(x_\tau y_\xi - y_\tau x_\xi)\bar{q} - y_\xi \bar{f} + x_\xi \bar{g}\right] = 0 \qquad (7)$$

Replace $\frac{\partial}{\partial\tau}$, $\frac{\partial}{\partial\xi}$, $\frac{\partial}{\partial\eta}$ by some finite-difference operators, say

$$\frac{\partial}{\partial\tau} \leftarrow \Gamma_\tau \ , \ \frac{\partial}{\partial\xi} \leftarrow \Gamma_\xi \ , \ \frac{\partial}{\partial\eta} \leftarrow \Gamma_\eta$$

resulting in

$$\Gamma_\tau[I\bar{q}] + \Gamma_\xi[(y_\tau x^a_\eta - x_\tau y^a_\eta)\bar{q}] + \Gamma_\eta[(x_\tau y^a_\xi - y_\tau x^a_\xi)\bar{q}]$$

$$+ \Gamma_\xi[y^b_\eta \bar{f}] + \Gamma_\eta[-y^b_\xi \bar{f}] + \Gamma_\xi[-x^b_\eta \bar{g}] + \Gamma_\eta[x^b_\xi \bar{g}] = 0 \qquad 8$$

where the superscripts a and b appearing on the metric terms are used for later reference to identify two different numerical representations of the same quantity. Now for a uniform flow, since $\bar{f} = \bar{f}(\bar{q})$, $\bar{g} = \bar{g}(\bar{q})$, and it is required that $\bar{q}$ = constant throughout the flowfield for all time, then $\Gamma_\tau[\bar{q}] = \Gamma_\xi[\bar{q}] = \Gamma_\eta[\bar{q}] = \Gamma_\xi[\bar{f}] = \Gamma_\eta[\bar{f}] = \Gamma_\xi[\bar{g}] = \Gamma_\eta[\bar{g}] = 0$ and the following conditions result:

$$\Gamma_\tau[I] + \Gamma_\xi[y_\tau x^a_\eta - x_\tau y^a_\eta] + \Gamma_\eta[x_\tau y^a_\xi - y_\tau x^a_\xi] = 0 \qquad 9$$

$$\Gamma_\xi[y^b_\eta] - \Gamma_\eta[y^b_\xi] = 0 \qquad 10$$

$$\Gamma_\xi[x^b_\eta] - \Gamma_\eta[x^b_\xi] = 0 \qquad 11$$

Note that the differential analogs of these equations are simply <u>identities</u> for a well-behaved mapping. Clearly, Equations 10 and 11 are satisfied if the metrics are differenced with the same operators as those used in the finite-difference scheme. Equation 9, however, is a numerical representation of an identity coined by Thomas and

Lombard (20) as the geometric conservation law (GCL). It says that the GCL equation

$$\frac{\partial I}{\partial \tau} + \frac{\partial}{\partial \xi}\left[y_\tau x_\eta - x_\tau y_\eta\right] + \frac{\partial}{\partial \eta}\left[x_\tau y_\xi - y_\tau x_\xi\right] = 0 \qquad 12$$

must be differenced in an identical manner to the flow equations, Equation 7. This result is nontrivial only in the case of a time varying grid. Thomas and Lombard (20) reached the same conclusion with consistency arguments and analogy with finite-volume methods. It must be pointed out, however, that Equations 9-11 insofar as the present effort is concerned are strictly a result of choosing the strong conservation-law representation of the governing equations. If the weak conservation-law form of the equations had been chosen, a different set of conditions would result for the differencing of the metric gradients appearing in the source terms. The use of the nonconservative form of the equations results in no special geometry differencing requirements whatsoever.

Thus far two sets of metric values have been identified. They are referred to as a-metrics and b-metrics based on their superscript. The conditions indicated by Equations 10 and 11 dictate the manner in which the b-metrics are to be computed. However, the calculation of the a-metrics is still a free choice. Although the two sets represent the same physical quantities, they need not be numerically equivalent. In fact, it is shown in a subsequent section that the manner in which the a-metrics are computed is dictated by the accuracy with which the inte-

grated Jacobian value, I, resulting from Equation 9, represents the actual Jacobian of the mapping.

The calculation of the metrics requires knowledge of the coordinates (x,y) of each grid point. Determination of these coordinates and the speed with which the points move (x_τ,y_τ) is the subject of the next section.

Grid and Gridspeed Operators

In order to determine the metric quantities the coordinates (x,y) of each grid point must be known. Also grid point speeds (x_τ,y_τ) are required to advance both the flow solution and the Jacobian (Equations 7 and 12) in time. Due to the time varying nature of the grid boundaries the location and speed of each interior grid point is necessarily dependent upon the location and speed of the boundary points. Two methods for obtaining such dependence are now described. This elliptic-type dependence may be obtained for the grid by following the approach of Thompson, Thames, and Mastin (6). That is, given the boundary point coordinates (x,y), the interior grid point coordinates are required to satisfy the nonlinear elliptic coupled partial differential equations

$$G[x] = 0,\ G[y] = 0 \qquad 13$$

plus specified boundary values where

$$G = \alpha\frac{\partial^2}{\partial\xi^2} - 2\beta\frac{\partial^2}{\partial\xi\partial\eta} + \gamma\frac{\partial^2}{\partial\eta^2}$$

$$+ I^2\{P(\tau,\xi,\eta)\frac{\partial}{\partial\xi} + Q(\tau,\xi,\eta)\frac{\partial}{\partial\eta}\} \qquad 14$$

$$\alpha = x_\eta^2 + y_\eta^2 \qquad 15$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta \qquad 16$$

$$\gamma = x_\xi^2 + y_\xi^2 \qquad 17$$

and $P(\tau,\xi,\eta)$, $Q(\tau,\xi,\eta)$ are forcing functions which may be used to concentrate grid lines where they are most needed.[1]

The requirement that (x,y) satisfy Equation 13 plus boundary conditions allows determination of the grid. However, the grid speed values are still unknown on the interior points. The boundary point speeds are assumed known since these values typically represent the speed of shock points, etc., which are determined from the flow solution. Interior values for (x_τ,y_τ) could, of course, be obtained with backward finite-differences but this requires extra information at the initial data surface and may be inconsistent with the scheme used to advance the boundary point locations in time. In addition, the solution to Equation 13 must be iterative due to the nonlinearity of the operator, G. Another approach, and the one used in the present study, is to differentiate Equation 13 with respect to τ which, for a one-to-one

---

[1] $P(\tau,\xi,\eta)$, $Q(\tau,\xi,\eta)$ are taken to be zero in the present study.

mapping, yields the equation

$$S[\bar{Z}] = \bar{r} \qquad 18$$

where

$$\bar{Z} = (x_\tau, y_\tau)$$

$$\bar{r} = -I^2(P_\tau x_\xi + Q_\tau x_\eta \ , \ P_\tau y_\xi + Q_\tau y_\eta) \qquad 19$$

$$S = \begin{pmatrix} G + C_1(\tau,\xi,\eta)\frac{\partial}{\partial\xi} + C_2(\tau,\xi,\eta)\frac{\partial}{\partial\eta} & \vdots & C_3(\tau,\xi,\eta)\frac{\partial}{\partial\xi} + C_4(\tau,\xi,\eta)\frac{\partial}{\partial\eta} \\ \cdots\cdots\cdots & \vdots & \cdots\cdots\cdots \\ C_5(\tau,\xi,\eta)\frac{\partial}{\partial\xi} + C_6(\tau,\xi,\eta)\frac{\partial}{\partial\eta} & \vdots & G + C_7(\tau,\xi,\eta)\frac{\partial}{\partial\xi} + C_8(\tau,\xi,\eta)\frac{\partial}{\partial\eta} \end{pmatrix}$$

and

$$\begin{aligned}
C_1(\tau,\xi,\eta) &= 2(x_{\eta\eta}x_\xi - x_{\xi\eta}x_\eta + I(Px_\xi + Qx_\eta)y_\eta) \\
C_2(\tau,\xi,\eta) &= 2(x_{\xi\xi}x_\eta - x_{\xi\eta}x_\xi - I(Px_\xi + Qx_\eta)y_\xi) \\
C_3(\tau,\xi,\eta) &= 2(x_{\eta\eta}y_\xi - x_{\xi\eta}y_\eta - I(Px_\xi + Qx_\eta)x_\eta) \\
C_4(\tau,\xi,\eta) &= 2(x_{\xi\xi}y_\eta - x_{\xi\eta}y_\xi + I(Px_\xi + Qx_\eta)x_\xi) \\
C_5(\tau,\xi,\eta) &= 2(y_{\eta\eta}x_\xi - y_{\xi\eta}x_\eta + I(Py_\xi + Qy_\eta)y_\eta) \\
C_6(\tau,\xi,\eta) &= 2(y_{\xi\xi}x_\eta - y_{\xi\eta}x_\xi - I(Py_\xi + Qy_\eta)y_\xi) \\
C_7(\tau,\xi,\eta) &= 2(y_{\eta\eta}y_\xi - y_{\xi\eta}y_\eta - I(Py_\xi + Qy_\eta)x_\eta) \\
C_8(\tau,\xi,\eta) &= 2(y_{\xi\xi}y_\eta - y_{\xi\eta}y_\xi + I(Py_\xi + Qy_\eta)x_\xi)
\end{aligned} \qquad 20$$

Once the coordinates (x,y) are known at each grid point, the metric quantities and their derivatives may be obtained with finite-differences. The result is that the system of partial differential equations represented by Equation 18 is linear in the dependent variables (x_τ, y_τ) with known variable coefficients and a direct method of solution to its finite-difference representation may be employed to determine (x_τ, y_τ) at all interior points when given the boundary point speeds. In addition, the grid point locations may be determined from a simple time integration of these computed speeds rather than by solving the nonlinear system given by Equation 13. This point is discussed in the next chapter.

CHAPTER IV. COUPLING OF GEOMETRY TREATMENT AND FINITE-DIFFERENCE SCHEME

Several points are considered in this chapter involving the accuracy of the procedures developed. This accuracy is intimately connected to the coupling which exists between the finite-difference scheme chosen to integrate the governing equations and the treatment of the geometry. MacCormack's standard unsplit predictor-corrector scheme (21) is used to integrate the flow equations, Equation 7, and the GCL equation, Equation 12 in time. Define Δ_ℓ, ∇_ℓ to be forward and backward difference operators respectively on the ℓ-index ($\Delta_j[Q_{j,k}] = Q_{j+1,k} - Q_{j,k}$). Then the finite-difference form of Equations 7 and 12 becomes

predictors:

$$(I\bar{q})^{\overline{n+1}} = (I\bar{q})^n - \frac{\Delta\tau}{\Delta\xi}\Delta_j\left[\{(y_\tau x_\eta^a - x_\tau y_\eta^a)\bar{q} + y_\eta^b\bar{f} - x_\eta^b\bar{g}\}^n\right] - \frac{\Delta\tau}{\Delta\eta}\Delta_k\left[\{(x_\tau y_\xi^a - y_\tau x_\xi^a)\bar{q} - y_\xi^b\bar{f} + x_\xi^b\bar{g}\}^n\right] \qquad 21$$

$$I^{\overline{n+1}} = I^n - \frac{\Delta\tau}{\Delta\xi}\Delta_j\left[(y_\tau x_\eta^a - x_\tau y_\eta^a)^n\right] - \frac{\Delta\tau}{\Delta\eta}\Delta_k\left[(x_\tau y_\xi^a - y_\tau x_\xi^a)^n\right] \qquad 22$$

correctors:

$$(I\bar{q})^{n+1} = \frac{1}{2}\{(I\bar{q})^{\overline{n+1}} + (I\bar{q})^n - \frac{\Delta\tau}{\Delta\xi}\nabla_j\left[\{(y_\tau x_\eta^a - x_\tau y_\eta^a)\bar{q} + y_\eta^b\bar{f} - x_\eta^b\bar{g}\}^{\overline{n+1}}\right] - \frac{\Delta\tau}{\Delta\eta}\nabla_k\left[\{(x_\tau y_\xi^a - y_\tau x_\xi^a)\bar{q} - y_\xi^b\bar{f} + x_\xi^b\bar{g}\}^{\overline{n+1}}\right]\} \qquad 23$$

$$I^{n+1} = \frac{1}{2}\{I^{\overline{n+1}} + I^{n} - \frac{\Delta\tau}{\Delta\xi}\nabla_j\left[(y_\tau x^a_\eta - x_\tau y^a_\eta)^{\overline{n+1}}\right] - \frac{\Delta\tau}{\Delta\eta}\nabla_k\left[(x_\tau y^a_\xi - y_\tau x^a_\xi)^{\overline{n+1}}\right]\} \qquad 24$$

The value of I^{n+1} obtained from Equation 24 must accurately reflect the true grid structure at n+1. The grid must therefore be advanced in time in such a way that the Jacobian, I_g^{n+1}, as computed with finite-differences from the new grid point locations $(x,y)^{n+1}$ is an accurate representation of I^{n+1}. That is

$$E = \| I_g^{n+1} - I^{n+1} \| \leq \epsilon$$

where ϵ is a small number. Two methods exist for controlling the order of ϵ. The manner in which the new grid is obtained from a time integration controls the value of I_g^{n+1} and the manner in which the a-metrics in Equations 21-24 are treated controls the value of I^{n+1}. It is shown in the Appendix that if the grid is advanced in time by the Euler predictor-modified Euler corrector scheme given by

predictors:

$$x^{\overline{n+1}} = x^n + \Delta\tau\, x^n_\tau$$

$$y^{\overline{n+1}} = y^n + \Delta\tau\, y^n_\tau \qquad 25$$

correctors:

$$x^{n+1} = \frac{1}{2}(x^n + x^{\overline{n+1}} + \Delta\tau\, x^{\overline{n+1}}_\tau)$$

$$y^{n+1} = \frac{1}{2}(y^n + y^{\overline{n+1}} + \Delta\tau\, y_\tau^{\overline{n+1}}) \qquad 26$$

and the a-metrics are differenced (with $\Delta\xi = \Delta\eta = 1$) as

$$\begin{aligned} x_\eta^{a^n} &= \delta_k S_j^-[x^n]\ ,\ y_\eta^{a^n} = \delta_k S_j^-[y^n] \\ x_\xi^{a^n} &= \delta_j S_k^-[x^n]\ ,\ y_\xi^{a^n} = \delta_j S_k^-[y^n] \end{aligned} \qquad 27$$

$$\begin{aligned} x_\eta^{a^{\overline{n+1}}} &= \delta_k S_j^+[x^{\overline{n+1}}]\ ,\ y_\eta^{a^{\overline{n+1}}} = \delta_k S_j^+[y^{\overline{n+1}}] \\ x_\xi^{a^{\overline{n+1}}} &= \delta_j S_k^+[x^{\overline{n+1}}]\ ,\ y_\xi^{a^{\overline{n+1}}} = \delta_j S_k^+[y^{\overline{n+1}}] \end{aligned} \qquad 28$$

where δ_ℓ is the central difference operator on the ℓ-index {i.e., $\delta_\ell[Q_{\ell,m}] = \frac{1}{2}(Q_{\ell+1,m} - Q_{\ell-1,m})$} and $S_\ell^\pm$ is the shift operator on the ℓ-index where the superscript $\pm$ indicates the shift direction {i.e., $S_\ell^+[Q_{\ell,m}] = Q_{\ell+1,m}$}, then

$$\varepsilon = O(\Delta\tau^3, \Delta\tau^2\Delta\xi, \Delta\tau^2\Delta\eta)$$

This is of higher order than the accuracy of the numerical determination of I_g^{n+1} from the formula

$$I_g^{n+1} = \delta_j[x^{n+1}]\delta_k[y^{n+1}] - \delta_j[y^{n+1}]\delta_k[x^{n+1}] \qquad 29$$

An additional result of obtaining $(x,y)^{n+1}$ from Equations 25 and 26 is that

$$\| G^{n+1}[x^{n+1}] \| = \varepsilon_1\ ,\ \| G^{n+1}[y^{n+1}] \| = \varepsilon_2$$

where $\varepsilon_{1,2} \sim O(\Delta\tau^3)$. That is, (x,y) at the new time are very accurate representations of the values obtained by solving Equation 13 at n+1. This fact provides some assurance that the grid will remain nicely structured as it moves in time.

CHAPTER V. ALGORITHM

The algorithm for applying the procedures described in the preceding chapters to an arbitrary module is described in what follows. A priori knowledge of the initial boundary point locations and their speeds, and the initial flow solution, $\overline{q}^n$, is assumed.

1. Given $(x,y)^n$ at all boundary points, compute $(x,y)^n$ at all interior points initially by solving the coupled equations $G^n[x] = 0$, $G^n[y] = 0$.

2. Compute the a-metrics from Equation 27.

3. Compute the b-metrics from

$$y_\eta^{b^n} = \Delta_k[y^n],\ x_\eta^{b^n} = \Delta_k[x^n]$$

$$y_\xi^{b^n} = \Delta_j[y^n],\ x_\xi^{b^n} = \Delta_j[x^n]$$

4. Given $(x_\tau,y_\tau)^n$ at all boundary points, compute $(x_\tau,y_\tau)^n$ at all interior points by solving Equation 18.

5. Compute the Jacobian, I^n, from the equation

$$I^n = I_g^n = \delta_j[x^n]\delta_k[y^n] - \delta_k[x^n]\delta_j[y^n]$$

6. Apply Equations 21 and 22 to yield $(I\overline{q})^{\overline{n+1}}$ and $I^{\overline{n+1}}$ at all grid points.[1]

---

[1]Special difference equations are actually used at boundary points.

7. Apply Equation 25 to yield $(x,y)^{\overline{n+1}}$ at all grid points.

8. Compute $\overline{q}^{\overline{n+1}} = (I\overline{q})^{\overline{n+1}}/I^{\overline{n+1}}$ and apply boundary conditions to obtain the boundary speed $(x_\tau,y_\tau)^{\overline{n+1}}$ (i.e., shock speed, etc.).

9. Compute a-metrics from Equation 28.

10. Compute b-metrics from

$$y_\eta^{b\,\overline{n+1}} = \nabla_k\left[y^{\overline{n+1}}\right],\quad x_\eta^{b\,\overline{n+1}} = \nabla_k\left[x^{\overline{n+1}}\right]$$
$$y_\xi^{b\,\overline{n+1}} = \nabla_j\left[y^{\overline{n+1}}\right],\quad x_\xi^{b\,\overline{n+1}} = \nabla_j\left[x^{\overline{n+1}}\right]$$

11. Given $(x_\tau,y_\tau)^{\overline{n+1}}$ at all boundary points, compute $(x_\tau,y_\tau)^{\overline{n+1}}$ at all interior points by solving Equation 18.

12. Apply Equations 23 and 24 to yield $(I\overline{q})^{n+1}$, I^{n+1} at all grid points.[1]

13. Apply Equation 26 to yield $(x,y)^{n+1}$ at all grid points.

14. Compute $\overline{q}^{n+1} = (I\overline{q})^{n+1}/I^{n+1}$ and apply boundary conditions to correct $\overline{q}^{n+1}$ and provide $(x_\tau,y_\tau)^{n+1}$ on the boundaries.

15. Go to step 2.

---

[1]Special difference equations are actually used at boundary points.

CHAPTER VI. INITIAL AND BOUNDARY CONDITIONS

Initial Conditions

The techniques developed in the present effort revolve around the concept of generality. The determination of an initial flow field for an arbitrary problem does not lend itself to a general treatment. As a result, the initial solution for a given flow configuration is treated as an independent problem and is not part of the existing computer code.

Boundary Conditions

The algorithm described in the previous chapter applies to an arbitrary module. Steps 8 and 14 require the application of some boundary condition procedures. It is through this application that neighboring modules are coupled together, flow tangency requirements are satisfied, etc. The boundary conditions used in the present effort are all applied with the same general philosophy. That is, some approximate boundary values of the flow variables are obtained at all boundaries of all modules by advancing the flow solution at the boundary points in time with modified versions of Equations 21-24. These modified versions incorporate one-sided differences where required. Once an estimate for the flow variables at the boundaries is obtained, it is corrected by some means to reflect the particular type of boundary present.

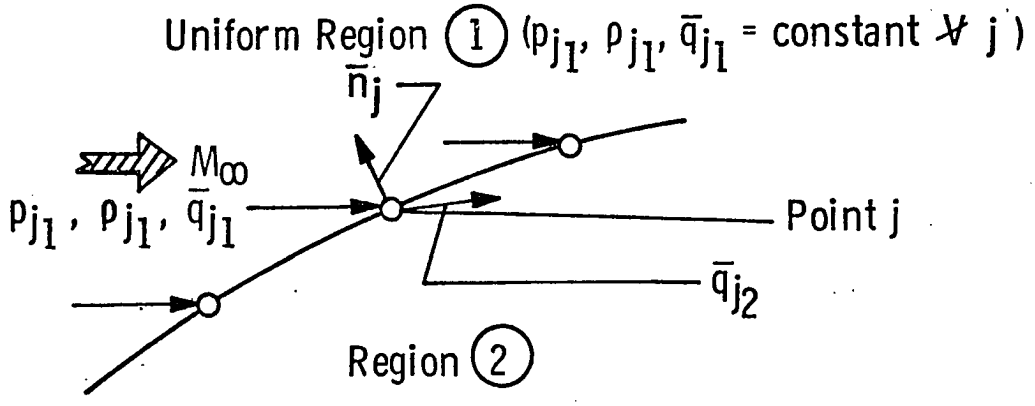
Body boundary conditions

Many procedures exist for applying flow tangency at a body surface (22,23). However, since this boundary type is not shared by another module, the boundary condition here is relatively uninteresting in the present context and will not be discussed at any length. The two approaches found to be most useful and simple to apply are mentioned briefly.

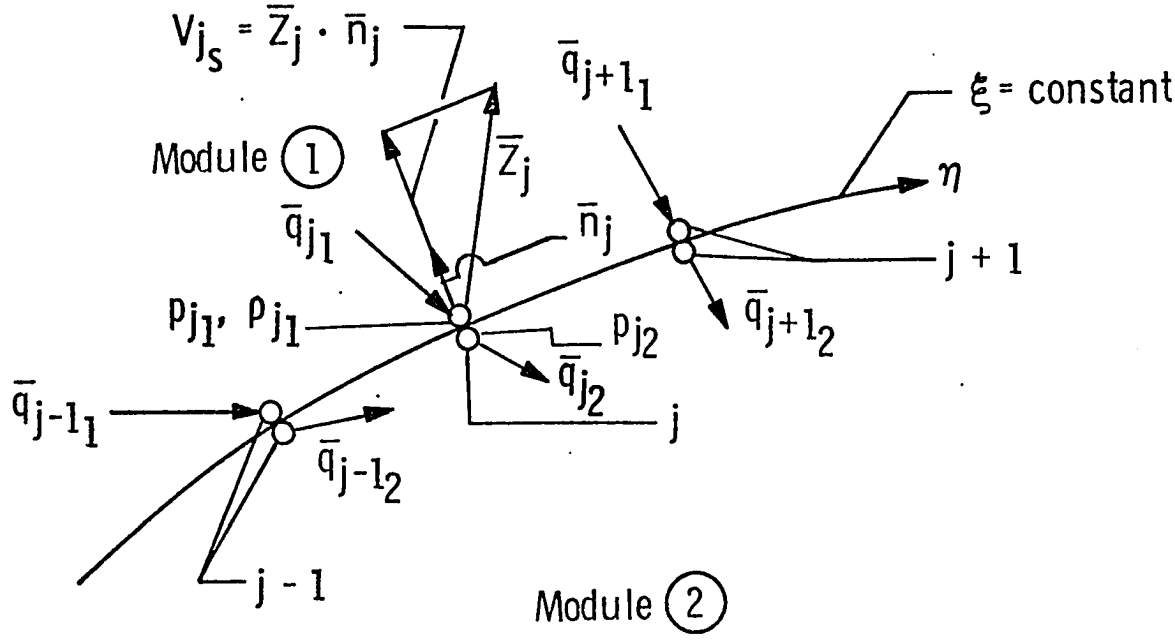
One approach is to accept the estimates of total momentum and pressure with an option to accept density or determine it from a known entropy value. The momentum components are then adjusted so as to provide the correct tangency condition. The second approach is to solve the surface normal momentum equation for pressure with the other variables treated as in the first approach.

Shock boundary conditions

A shock may or may not be an interface between two modules. Consider Figure 4. If the upstream side of the shock is a uniform flow as in Figure 4a, it need not be distinct and treated as a separate module. The more general case, shown in Figure 4b, will be discussed where two modules share the shock as a common boundary. The former situation is a special case of this. The Thomas et al. (24) pressure approach is used for obtaining the shock boundary speed normal to the current shock at, say, the jth shock point. This approach requires that the flow solution on the upstream side of the shock (module 1) and the pressure on the



a. Noninterface Shock Boundary



b. Interface Shock Boundary

Figure 4. Shock boundaries

downstream side (module 2) be known. These values are obtained when the flow equations are advanced in time at all the module boundaries. Once this information is known, the normal shock speed at the jth point is obtained from the equation

$$V_{j_s} = \bar{q}_{j_1} \cdot \bar{n}_j + \sqrt{\frac{p_{j_2}(\gamma + 1) + p_{j_1}(\gamma - 1)}{2\rho_{j_1}}}$$

The remaining flow variables on the downstream side are obtained from the Rankine-Hugoniot relations (25). Details of the application of Thomas' approach are given by Kutler (26).

In many cases it is undesirable to propagate a given shock point in a direction normal to the current shock. Thus the actual shock point speed $(x_\tau, y_\tau)_j$ is not necessarily equivalent to the normal shock speed just described. Many investigators propagate shock points along constant coordinate lines. This is also undesirable in some cases. The precise determination of the shock point speed from the normal speed and an additional condition is described in a later section and is applicable to the general module boundary point speed.

Slip surface boundary conditions

A slip surface boundary has no upstream or downstream side since the flow tangency condition is required. Consider Figure 5. Estimates for all the flow variables at the jth point in both modules 1 and 2 are obtained as described at the beginning of this chapter. These es-

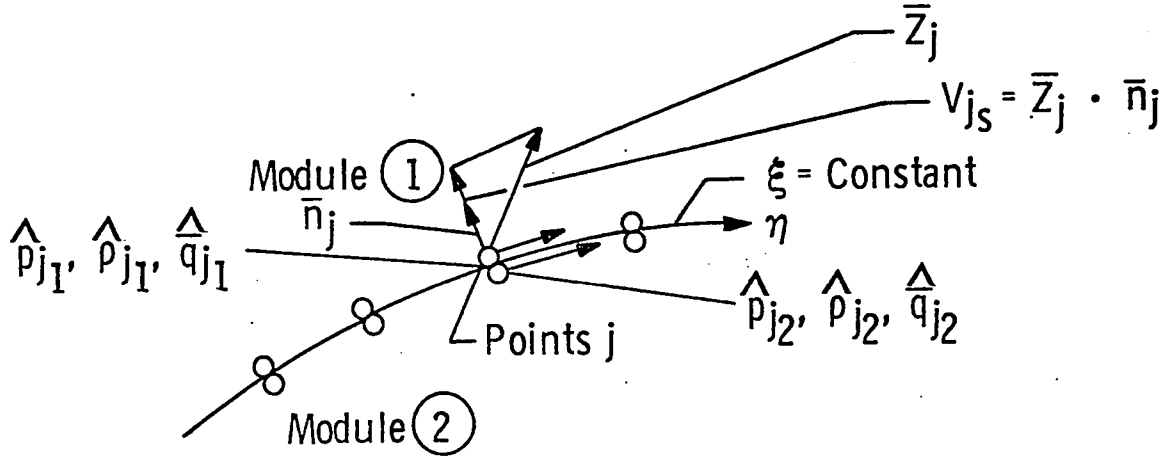


Figure 5. Slip surface interface

timates are denoted by

$$\hat{p}_{j_1},\ \hat{\rho}_{j_1},\ \hat{\bar{q}}_{j_1} = (\hat{u}_{j_1},\hat{v}_{j_1})$$

and

$$\hat{p}_{j_2},\ \hat{\rho}_{j_2},\ \hat{\bar{q}}_{j_2} = (\hat{u}_{j_2},\hat{v}_{j_2})$$

The normal unit vector to the current slip surface at point j is given by

$$\bar{n}_j = (\alpha,\beta)$$

where

$$\alpha = \frac{-y_\eta}{\sqrt{x_\eta^2 + y_\eta^2}},\quad \beta = \frac{x_\eta}{\sqrt{x_\eta^2 + y_\eta^2}}$$

in terms of the metrics at point j.[1] The correct solution at point j must satisfy the relations

$$p_{j_1} = p_{j_2}$$

$$\bar{q}_{j_1} \cdot \bar{n}_j = \bar{q}_{j_2} \cdot \bar{n}_j = V_{j_s} = \bar{Z}_j \cdot \bar{n}_j$$

where V_{j_s} is the velocity of the slip surface in the direction of $\bar{n}_j$. The procedure used to satisfy these conditions is now given.

---

[1] This assumes that the slip surface is given by ξ = constant.

Set

$$p_{j_1} = p_{j_2} = \frac{1}{2}(\hat{p}_{j_1} + \hat{p}_{j_2})$$

$$\bar{\hat{q}}_{j_1} = \hat{\hat{q}}_{j_1} + \frac{1}{2}(\hat{\hat{q}}_{j_2} \cdot \hat{\hat{n}}_j - \hat{\hat{q}}_{j_1} \cdot \hat{\hat{n}}_j)\hat{\hat{n}}_j$$

$$\bar{\hat{q}}_{j_2} = \hat{\hat{q}}_{j_2} - \frac{1}{2}(\hat{\hat{q}}_{j_2} \cdot \hat{\hat{n}}_j - \hat{\hat{q}}_{j_1} \cdot \hat{\hat{n}}_j)\hat{\hat{n}}_j$$

and

$$\rho_{j_1} = a\,\hat{\rho}_{j_1} + (1 - a)\left(\frac{p_{j_1}}{S_1}\right)^{\frac{1}{\gamma}}$$

$$\rho_{j_2} = b\,\hat{\rho}_{j_2} + (1 - b)\left(\frac{p_{j_2}}{S_2}\right)^{\frac{1}{\gamma}}$$

where a and b are 0 or 1 providing for the option to use the known entropy value, S_1 or S_2, at the upstream end of the slip surface boundary when computing the density. The actual determination of the slip surface point speed (x_τ, y_τ) is the same as for any other boundary point once the normal velocity, V_{j_s}, of the point is determined. As mentioned for the shock boundary condition, this is discussed in a more general setting in a later section. Other methods for treating slip surface boundaries are given by Shankar, Anderson, and Kutler (27), de Neef (28), and Marconi (29).

Other boundary conditions

Various other types of boundaries often arise such as supersonic inflow, outflow, and symmetry plane boundaries. These boundaries are artificial in nature and do not act as interface boundaries between separate modules. Their treatment is straightforward and therefore will not be discussed here.

Corner Conditions

The boundary conditions previously described work quite well for interior[1] boundary points. Difficulties arise, however, at points where two boundaries intersect. At such points, typically at least two sets of conditions must be met by the solution. The problem is over-specified and the correct answer satisfying all sets of conditions is obtained only as the steady-state limiting solution if such a solution is obtainable. The result is that only some of the conditions may be specifically enforced as boundary conditions and the remainder of them are used to check the validity of the resulting numerical solution. Such a procedure is used by Ludloff and Friedman (5) on a planar shock diffraction problem. The real trick is to determine which conditions are most important.

---

[1]Boundary points j = 2,3,••••,JMAX-1 are considered as interior boundary points.

Many different types of corners may arise in a general setting such as the present one. In fact, if all possibilities are accounted for and n different types of boundaries are permitted, then m different types of corners can arise where m is given by

$$m = \sum_{k=1}^{n} k$$

Thus, if only five boundary types are permitted, say, shock, slip surface, body, supersonic outflow, and supersonic inflow, then fifteen corner types are possible. Two representative situations are described in the following two subsections.

Simple intersection

Consider the intersection corner of a supersonic outflow boundary and a body as depicted in Figure 6. The flow solution at points B and C is determined by extrapolation along lines 2 and 3. The flow solution at points D and E is obtained by enforcing surface tangency to the initial integrated estimates. The solution at point A is some combination of these two requirements. The actual procedure used is to first alter the estimate already present at point A by extrapolation along line 1 (the body surface). Then enforce surface tangency to the extrapolated solution.

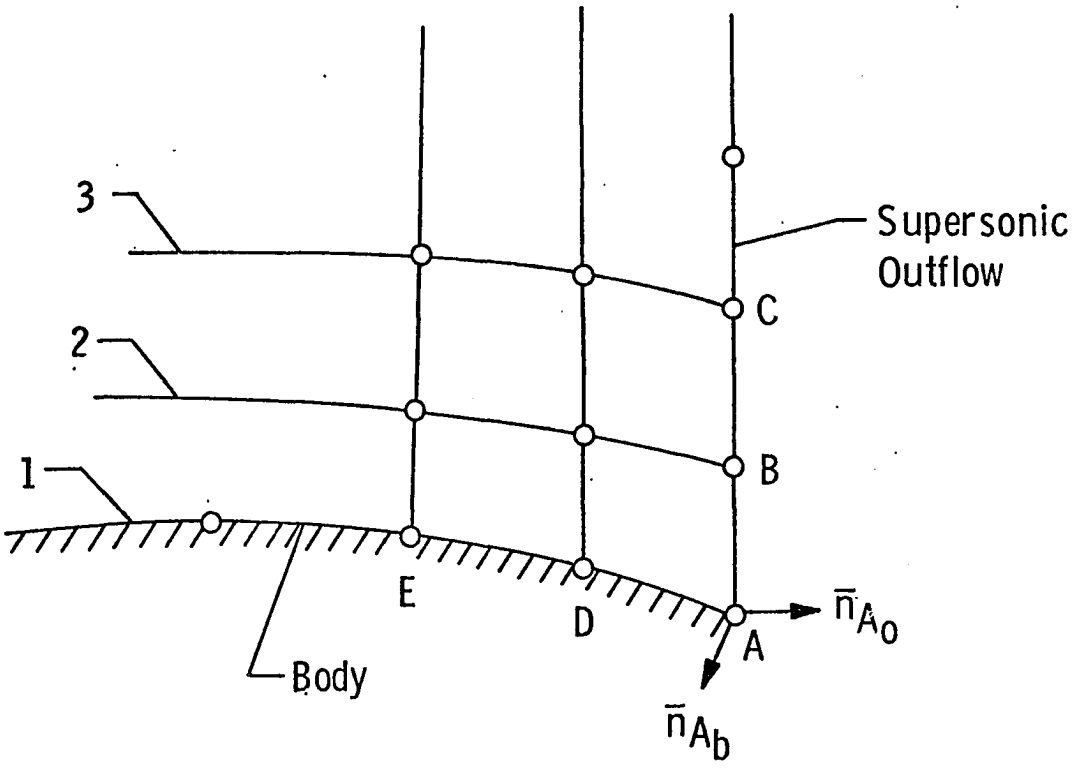


Figure 6. Simple intersection

Complex intersection

The most difficult corners to treat are those occurring at the ends of module interface boundaries. An example is the corner formed by the intersection of two shocks with a slip surface emanating from the intersection point. This multiple module common corner is shown in Figure 7. Four separate flow regions are actually involved here but for the sake of simplicity, assume that regions A and B are uniform flow regions separated by the straight shock I. Then point 'c' in module C and point 'd' in module D are treated as described in the shock boundary condition subsection. The points labeled 'a' are treated as described in the slip surface boundary condition subsection. The points labeled 'b' must somehow reflect both the fact that they lie at a shock-shock intersection point and that a slip surface is present with additional conditions to be satisfied. Satisfaction of all conditions is only met once the shocks and the slip surface are in the correct location and satisfy the correct jump conditions. The procedure used in the present study is as follows. Compute one value of pressure at points 'b' from the average

$$p_1 = p_2 = p_b = \frac{1}{2}(\hat{p}_1 + \hat{p}_2)$$

where the hat quantities are integrated estimates of pressure at points 1 and 2. Use the value of p_b and the known upstream flow variables in regions A and B to apply the shock scheme and provide the flow variables on the downstream side at points 1 and 2. This solution satis-

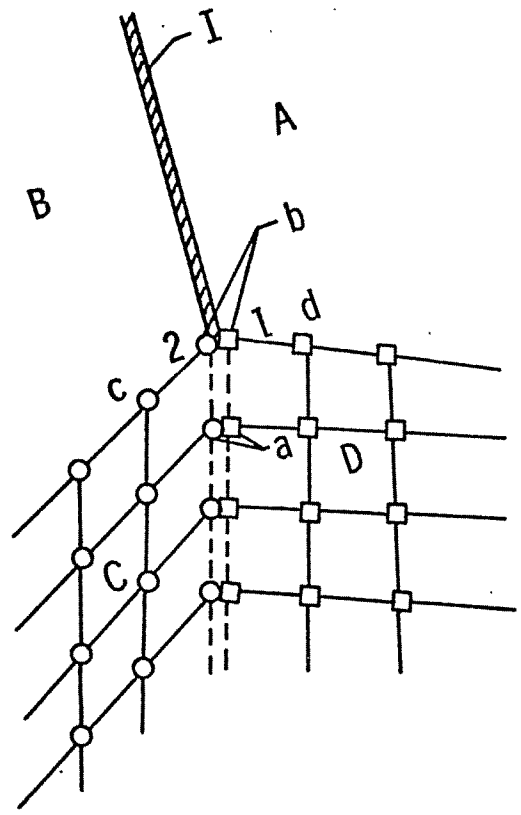


Figure 7. Complex intersection

fies the shock relations and the requirement that $p_1 = p_2$ but the slip surface tangency requirement is not satisfied and will not be in general until the solution converges. If the slip surface tangency condition is now enforced, then the shock relations are no longer satisfied. Since the slip surface is an effect rather than a cause, it is reasonable to assume that the satisfaction of the shock relations is more important.

General Corner Speed

The grid points on the boundaries all have an associated normal grid speed component which is obtained by some method. This method has been described for shocks and slip surfaces. The points of intersection of two boundaries must therefore have a grid speed component normal to each boundary. Regardless of the type of boundary intersection, the corner grid point speed is determined as shown in Figure 8. Define

$$\bar{n}_A = (\alpha_A, \beta_A), \; \bar{n}_B = (\alpha_B, \beta_B)$$

$$V_{A_N} = \bar{V}_{A_N} \cdot \bar{n}_A, \; V_{B_N} = \bar{V}_{B_N} \cdot \bar{n}_B$$

Then the corner grid speed is given by

$$Z = \frac{1}{\alpha_A \beta_B - \alpha_B \beta_A} \left(\beta_B \, V_{A_N} - \beta_A \, V_{B_N}, \alpha_A \, V_{B_N} - \alpha_B \, V_{A_N} \right)$$

$$= (x_\tau, y_\tau) \qquad (30)$$

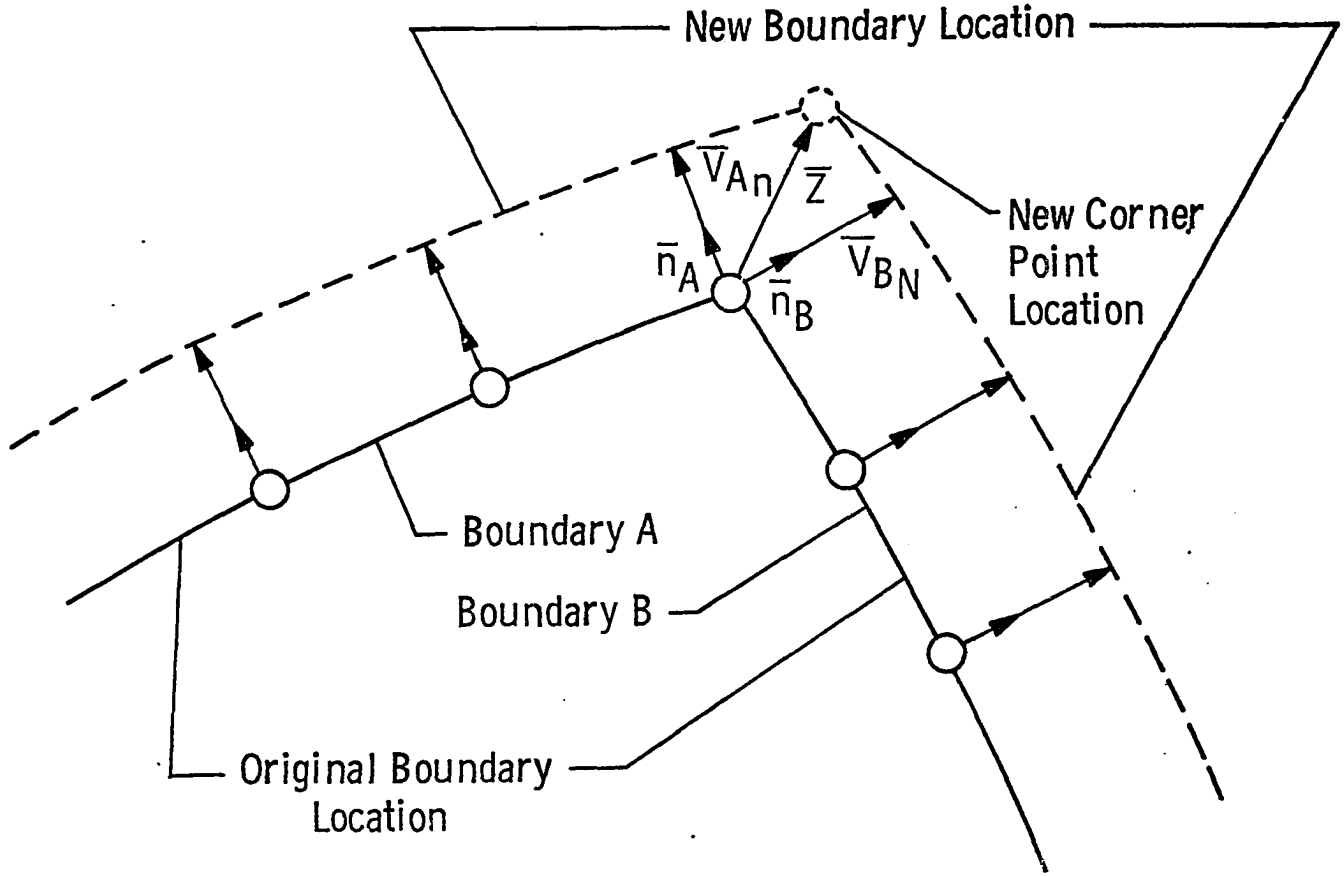


Figure 8. Corner grid point speeds

General Boundary Point Speed

Once the corner point speeds are determined at both ends of a module boundary and the boundary interior normal speeds are determined, then the actual evaluation of the boundary point speeds (x_τ, y_τ) may proceed (see Figure 9). The known quantities are $\bar{Z}_1$, $\bar{Z}_J$, and $\bar{V}_{j_N}$, $j = 1,2,\cdots,J$.

Compute ε_1, ε_J from

$$\varepsilon_1 = (\bar{Z}_1 - \bar{V}_{1_N}) \cdot (\bar{n}_1 \times \bar{k})$$

$$\varepsilon_J = (\bar{Z}_J - \bar{V}_{J_N}) \cdot (\bar{n}_J \times \bar{k})$$

where $\bar{k}$ is the unit vector normal to the x-y plane. Then for the jth interior boundary point,

$$\varepsilon_j = \varepsilon_1 + \frac{j - 1}{J - 1}(\varepsilon_J - \varepsilon_1) \qquad 31$$

and

$$\bar{\varepsilon}_j = \varepsilon_j(\bar{n}_j \times \bar{k})$$

so the speed of the jth boundary point is simply

$$\bar{Z}_j = \bar{V}_{j_N} + \bar{\varepsilon}_j \qquad 32$$

It may be that boundary point clustering toward some point on the boundary is required in which case something other than the linear relation

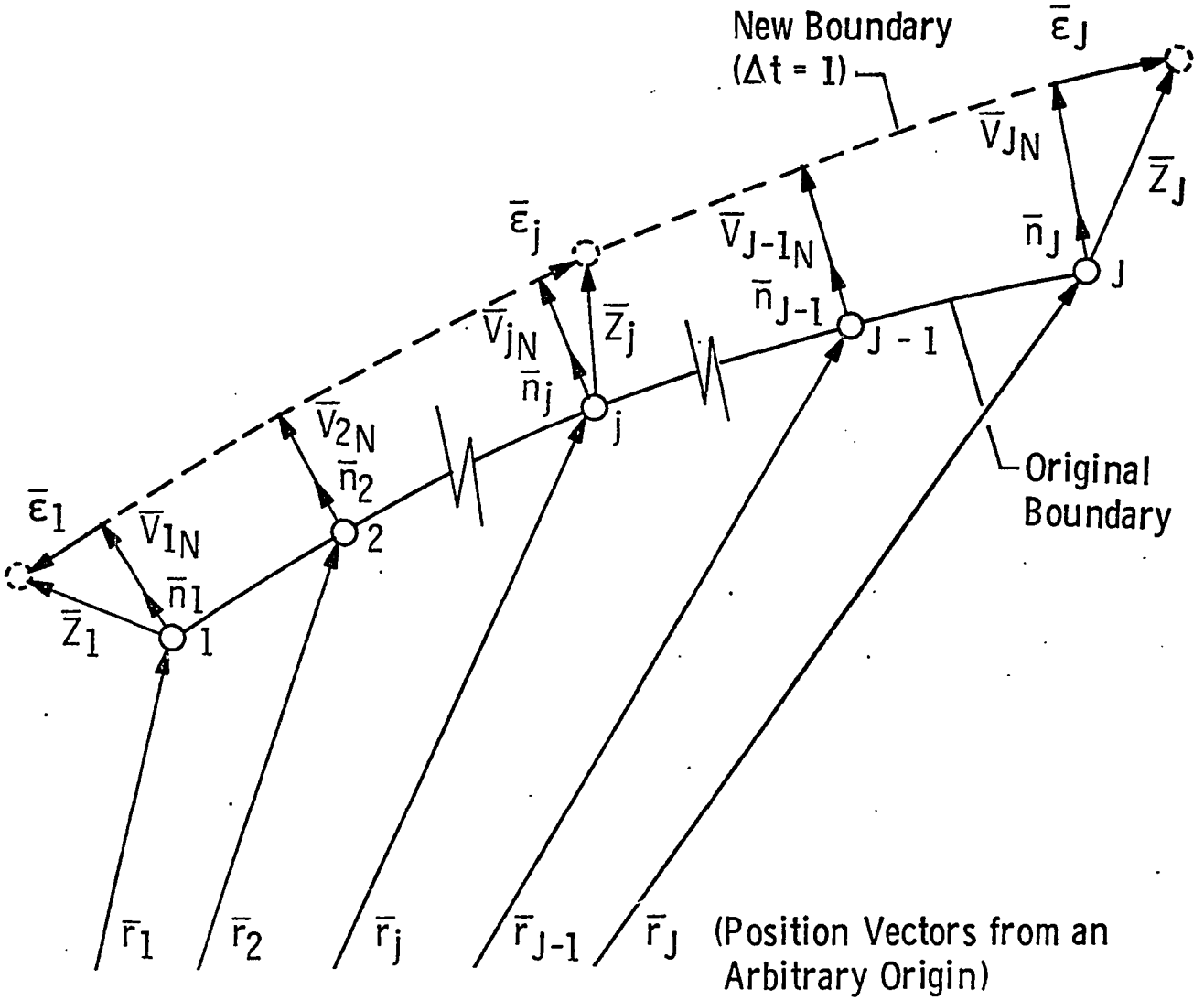


Figure 9. General boundary point speed

in Equation 31 could be chosen to compute ε_j. For example, define

$$d_1 = 0$$

$$d_j = |\bar{r}_j - \bar{r}_{j-1}|, \quad j = 2,3,4,\cdots,J$$

and

$$D = \sum_{j=1}^{J} d_j$$

Then

$$\varepsilon_j = \varepsilon_1 + r_j(\varepsilon_J - \varepsilon_1) \qquad 33$$

where

$$r_j = \frac{1}{D} \sum_{m=1}^{j} d_m$$

This procedure reduces to the original one when the d_j are all equal for $j > 1$. The advantage of the latter scheme is that the relative boundary point spacing will remain approximately the same as it was initially.

The procedures discussed for treating the problem geometry are capable of treating extremely general configurations. There are no built-in restrictions due to boundary description, base coordinate system choice, etc. Several interesting problems were computed with the procedures presented and are discussed in the next chapter.

CHAPTER VII. NUMERICAL RESULTS

This section consists of two parts. The first deals with problems which were solved with one module. These problems provide an excellent test of the new geometry procedures developed in this study. The second part deals with a simple preliminary test of the multiple module capability for a problem with two modules and one interface boundary.

A Single Module

Numerical results for several problems requiring a single module are presented. The first is the blunt body problem. A blunt body with a circular cross section and a free-stream Mach number of 4 was chosen as a test case. The converged grid is shown in Figure 10. The shock location is compared to the experimental results of Kim (30), and the empirical results of Billig (31). Both the shock location and the indicated sonic line location are compared to the numerical results of Daywitt and Anderson (32), Rizzi and Inouye (33), and Moretti (34). The comparisons are good between the present method and the various other numerical results. The stagnation streamline and body surface pressure is plotted in Figure 11. The slight overprediction of the stagnation point pressure is a result of using a two-dimensional Taylor expansion to arrive at the stagnation corner point solution. The known enthalpy and entropy values were not enforced but provide a convenient check of the numerical solution.

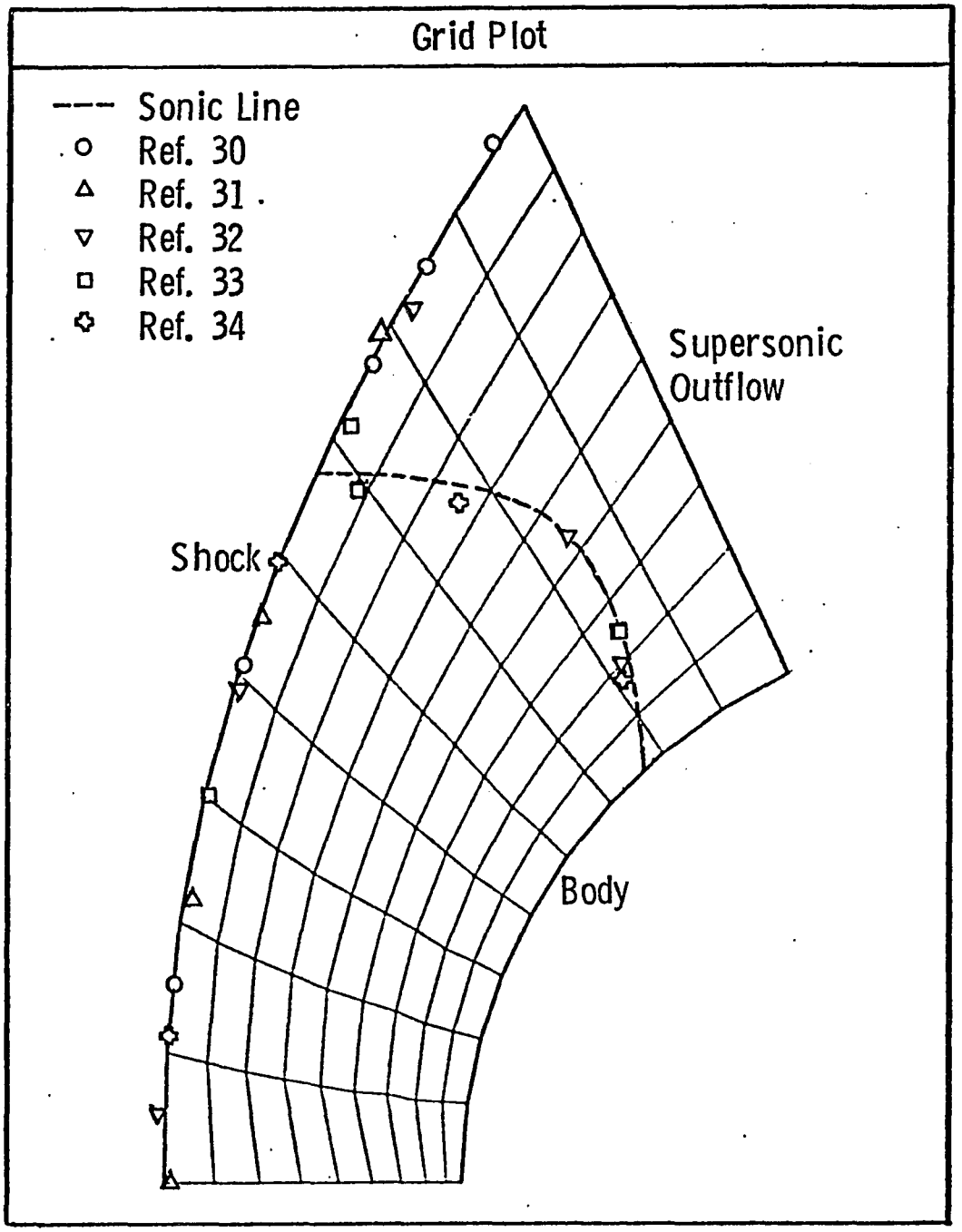


Figure 10. Cylindrical blunt body; Mach number = 4, body radius = 1

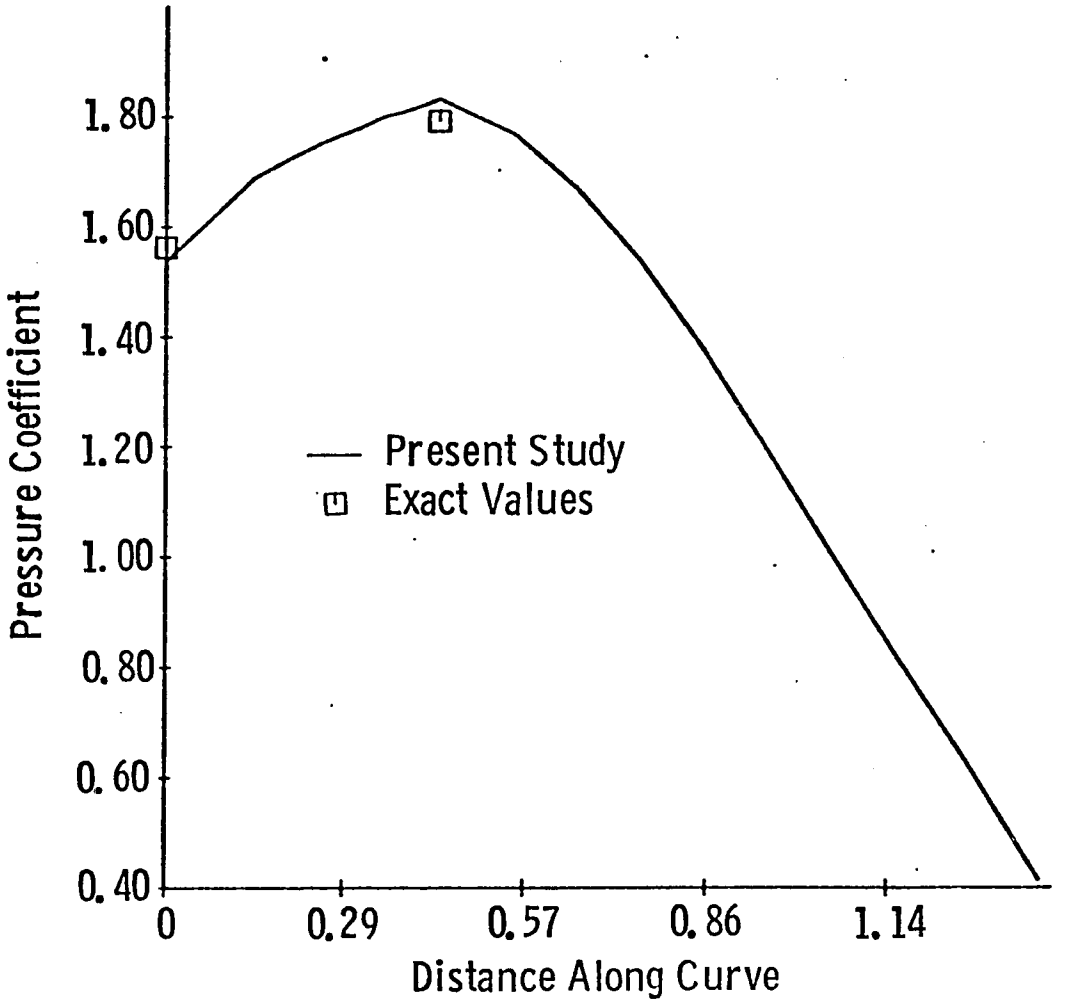


Figure 11. Cylindrical blunt body surface streamline pressure coefficient; Mach number = 4, body radius = 1

Figures 12 and 13 are results for a pointed 26.5 degree wedge at a free-stream Mach number of 1.44. This case has a detached shock wave as shown in Figure 12. The wedge body is gradually turned to be parallel to the free stream in order to obtain supersonic flow at the outflow boundary. Figure 12 illustrates the converged grid and sonic line and compares the present results to the experimental results of Griffith (35). Figure 13 depicts the stagnation streamline and body surface pressure distribution comparison. The results show only fair agreement with experiment. The discrepancy in the expansion region is primarily due to the fact that Griffith's results are for a body with a sharp rather than a rounded corner. An interesting feature of Figure 13 is the pressure profile near the stagnation point. This profile tends to spike rather than approach with zero slope as would be the case for a blunted leading edge. It is this large gradient region which is blamed for the discrepancy observed in the stagnation streamline pressure profile.

The results for a regular reflection planar shock diffraction problem with a shock Mach number of 4.71 and a ramp angle of 60 degrees are shown in Figures 14-16 and compared with the results of Kutler and Shankar (36) in Figures 15 and 16. Both the pressure and density distribution agree well with the Kutler and Shankar solution. It is interesting to note the behavior of the pressure profile for this problem as it approaches the stagnation point. The profile tends toward a zero

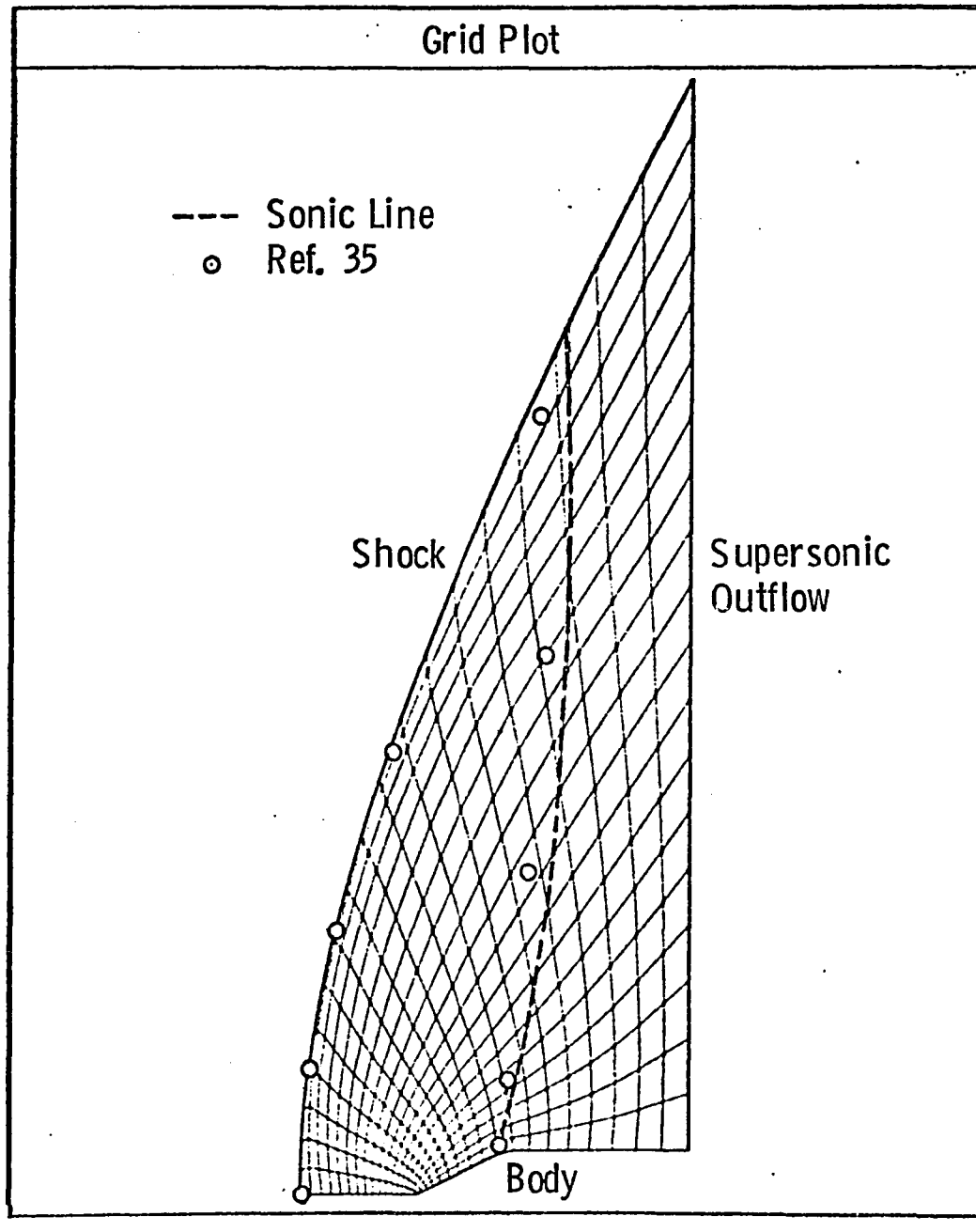


Figure 12. Pointed wedge with detached bow shock;
Mach number = 1.44, theta = 26.5 deg.

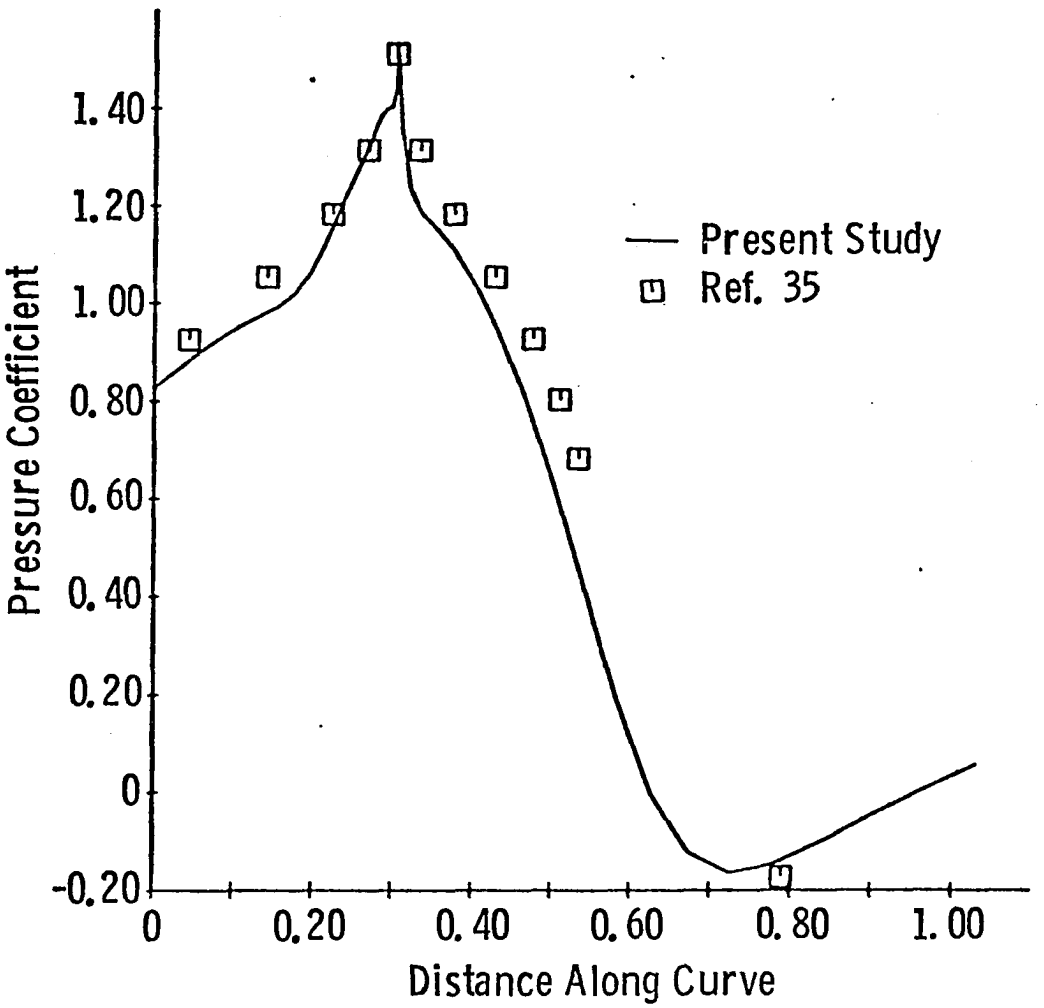


Figure 13. Pointed wedge with detached bow shock surface streamline pressure coefficient; Mach number = 1.44, theta = 26.5 degrees

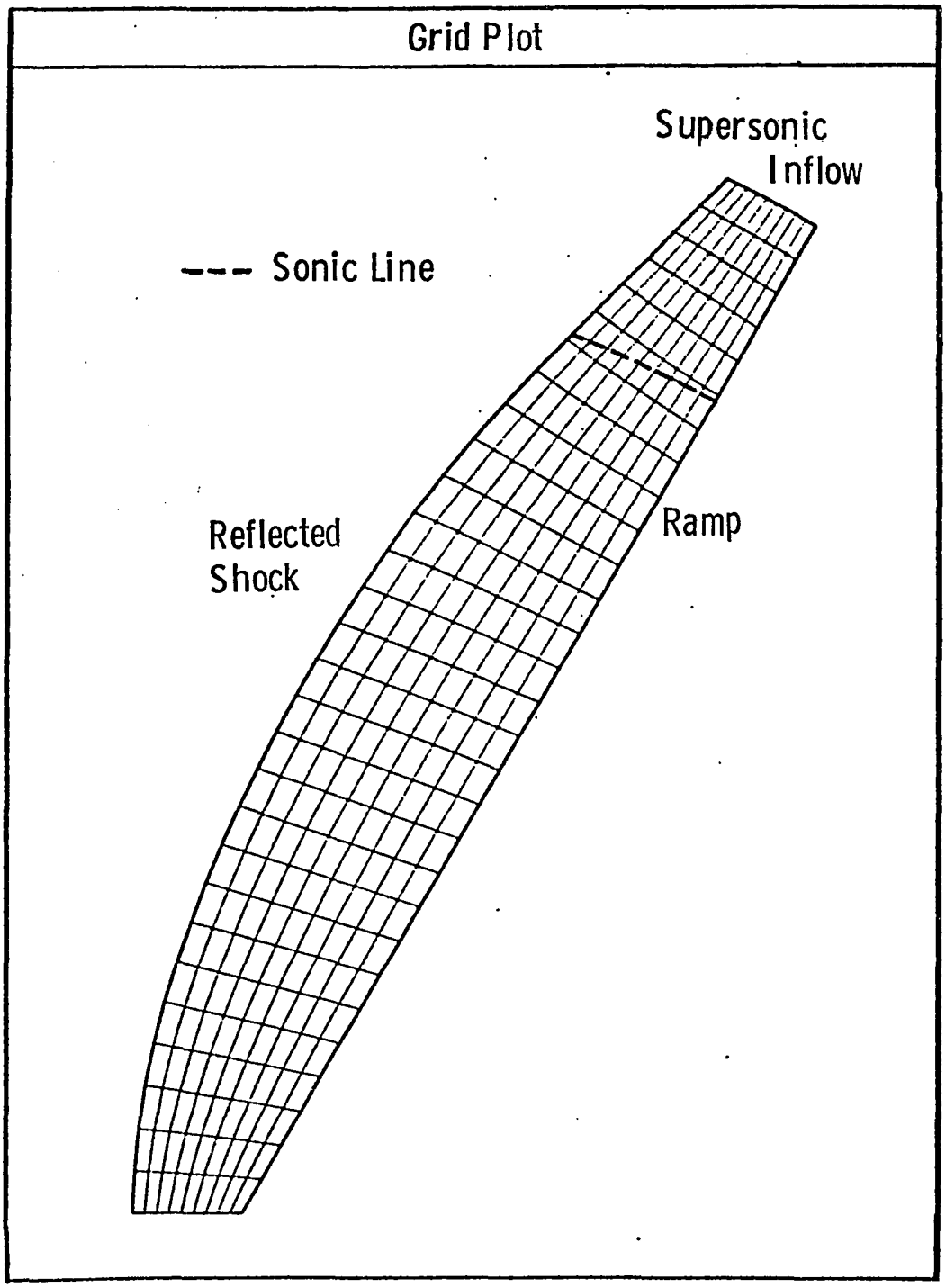


Figure 14. Regular reflection planar shock diffraction; ramp angle = 60 degrees, shock Mach number = 4.71

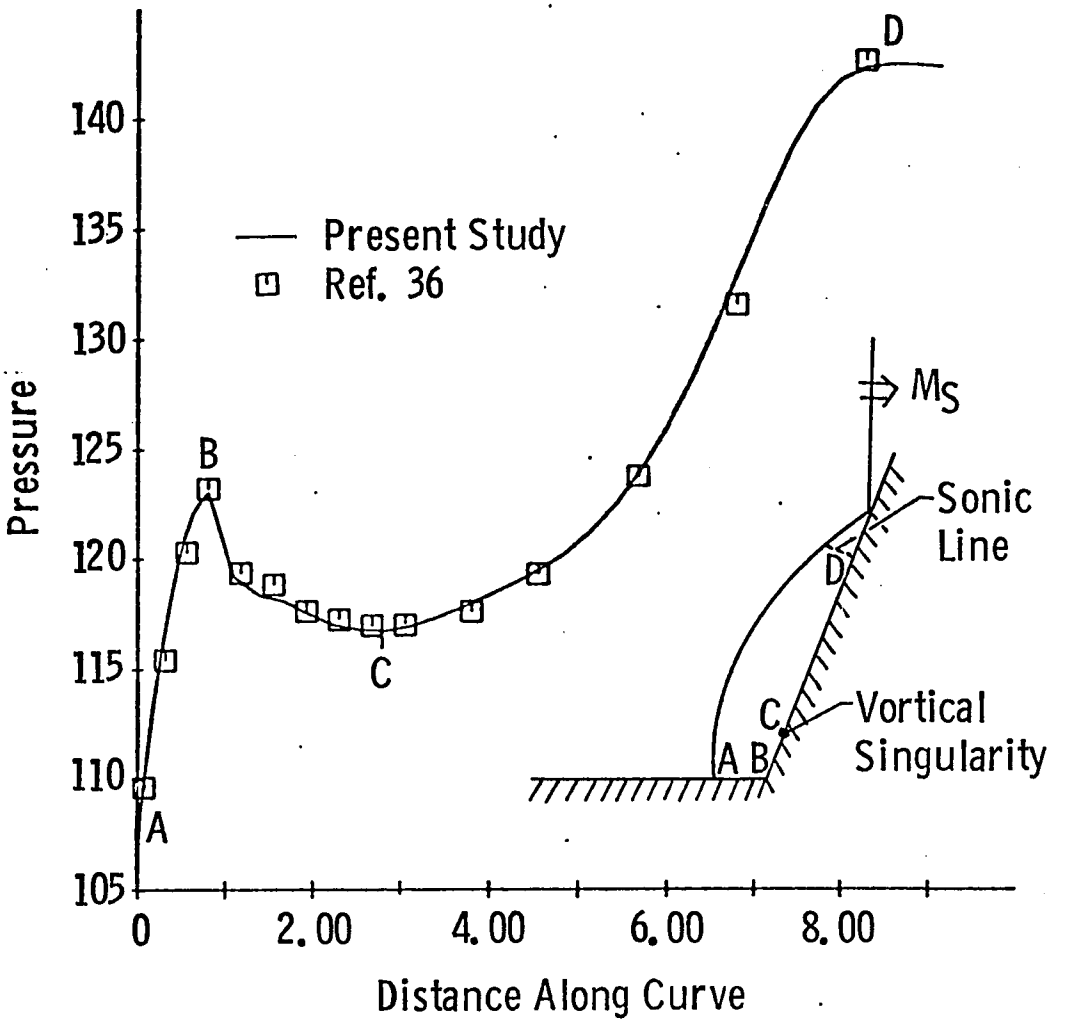


Figure 15. Regular reflection planar shock diffraction surface pressure; ramp angle = 60 degrees, shock Mach number = 4.71

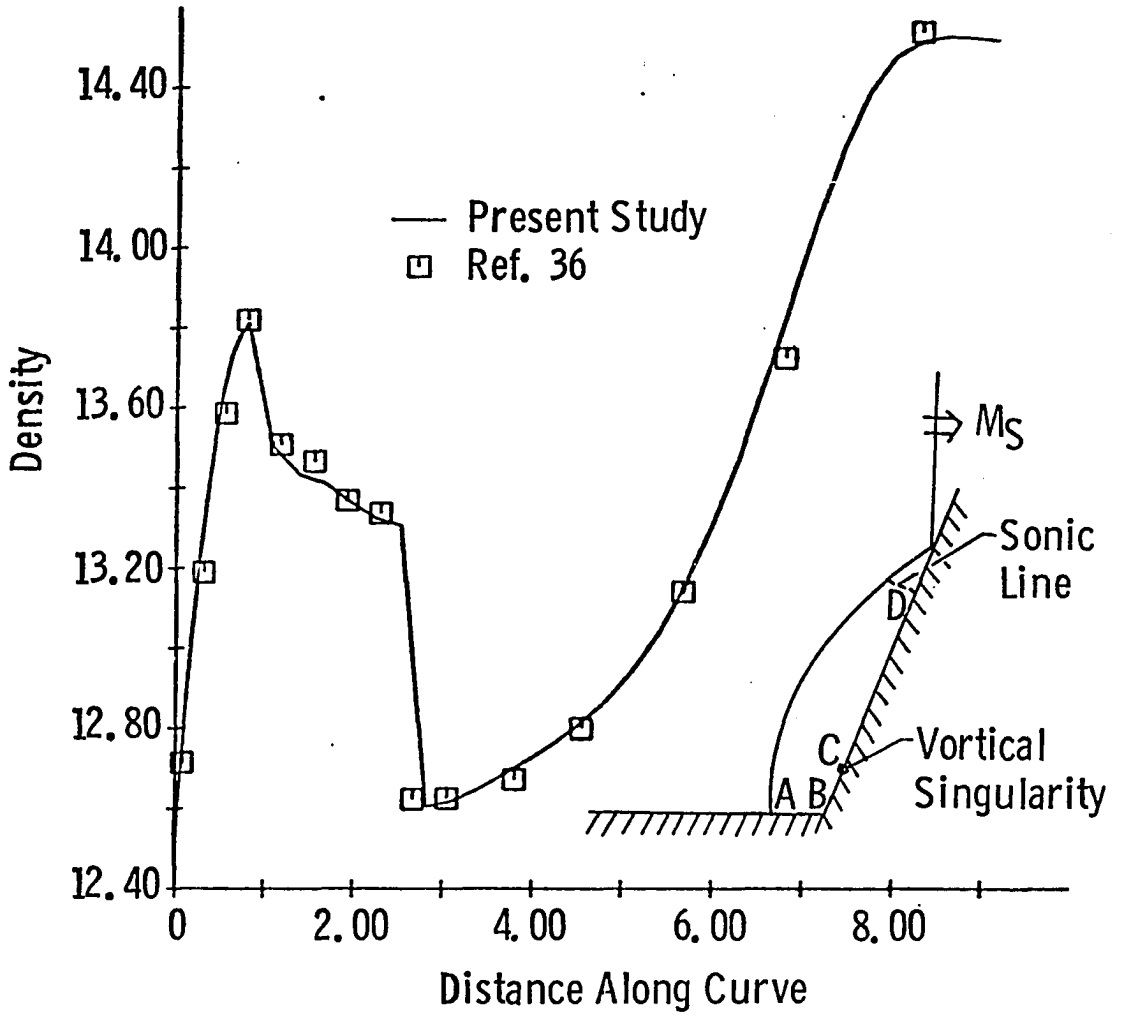


Figure 16. Regular reflection planar shock diffraction surface density; ramp angle = 60 degrees, shock Mach number = 4.71

slope when approaching along the stagnation streamline in contrast to the pointed wedge detached shock problem previously discussed but does tend to spike as the stagnation point is approached along the ramp. This problem is unique in that the grid never reaches a steady-state solution. As $\tau \to \infty$, $x_\tau \to x/\tau$ and $y_\tau \to y/\tau$ due to the self-similarity that exists with respect to time. The meaning of a converged grid in this case is when $\| x_\tau - x/\tau \|$ and $\| y_\tau - y/\tau \|$ are sufficiently small. Based on this definition, the converged grid is shown in Figure 14. Note that a vortical singularity exists for this problem and is clearly shown in Figure 16. This singularity was captured by the present finite-difference algorithm. The computational domain for this problem represents a four-sided physical region. The supersonic inflow side of this region is an artificial boundary and its validity is due to the two-dimensional flow region which exists between the sonic line, shown in Figure 14, and the shock ramp intersection point. If the numerical solution were required clear up to this intersection point, then the physical domain would have only three sides. The next example illustrates that such three-sided regions pose no serious threat to the present method.

Consider a ten percent thick pointed ogive body immersed in a Mach 2 free stream. The coverged grid is shown in Figure 17, thus illustrating a three-sided physical domain. This region is mapped into a computational four-sided region by introducing a 'corner' point along the leading edge shock. The Jacobian of the inverse transformation

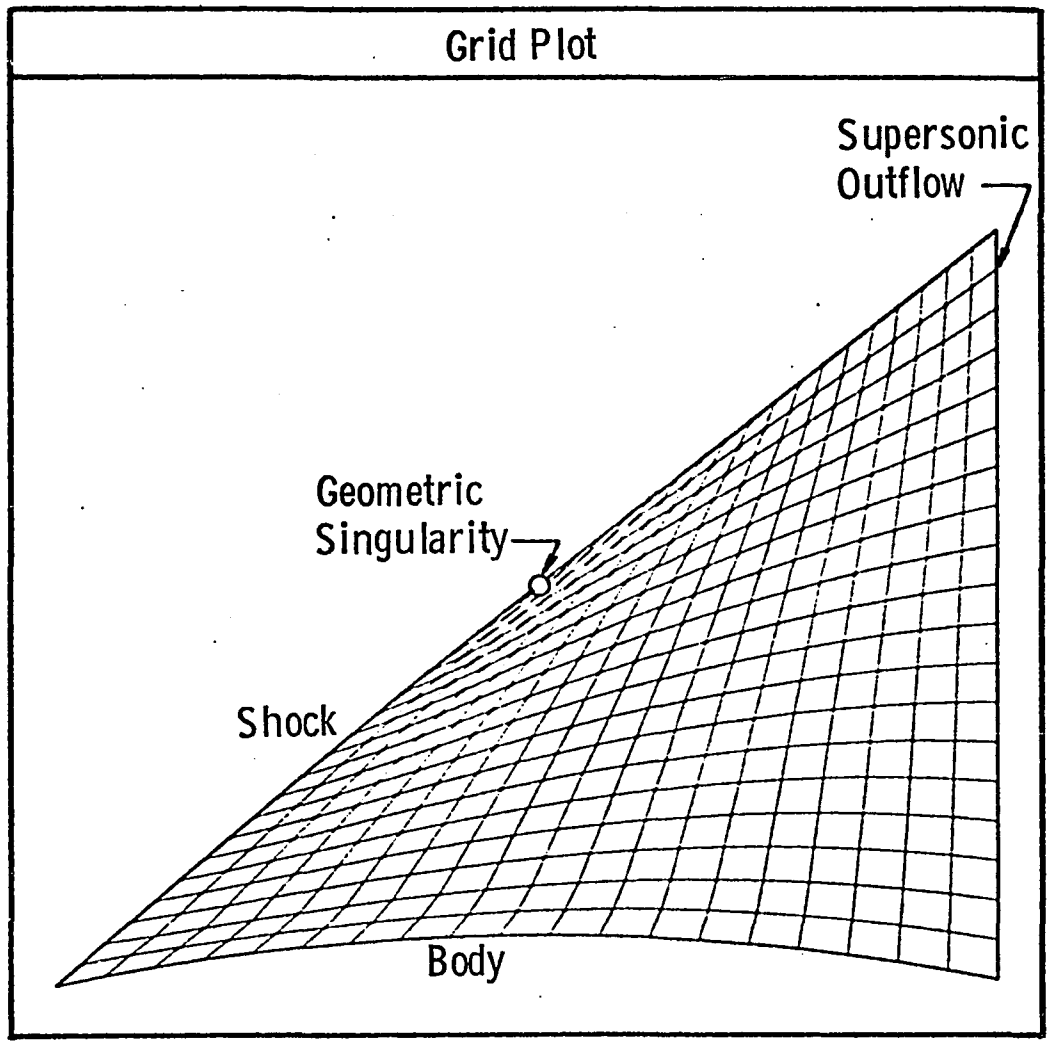


Figure 17. Single module ogive with geometric singularity on shock; Mach number = 2, thickness = 0.1

vanishes at this artificial corner thus creating a geometric singularity. This type of singularity is simply an additional corner point option in the existing computer code. Solutions were also obtained with this singularity introduced on the ogive surface and on the supersonic outflow boundary instead of on the shock. No difficulties were encountered with any of these cases. Figure 18 illustrates the ogive body surface pressure distribution along with the numerical results of Schiff (37). The comparison is good. Special attention must now be directed to the grid structure in the vicinity of the geometric singularity. The grid is extremely nonorthogonal in this region. Some researchers (38-39) have suggested in recent times that such nonorthogonality is undesirable from the standpoint of causing numerical difficulties or inaccuracies in the solution. This is certainly not true for the pointed ogive problem presented here.

Two Modules

The ogive body just discussed also serves as a test case of a double module problem with the trailing edge shock forming the interface boundary between the two modules. Figure 19 illustrates the converged grid and both the leading edge and trailing edge shocks. A comparison is made which shows good agreement with the results of the single module ogive problem. In this case, the aft region is three-sided and a geometric singularity is introduced along the symmetry boundary. The surface pressure distribution along the ogive and the symmetry boundary is

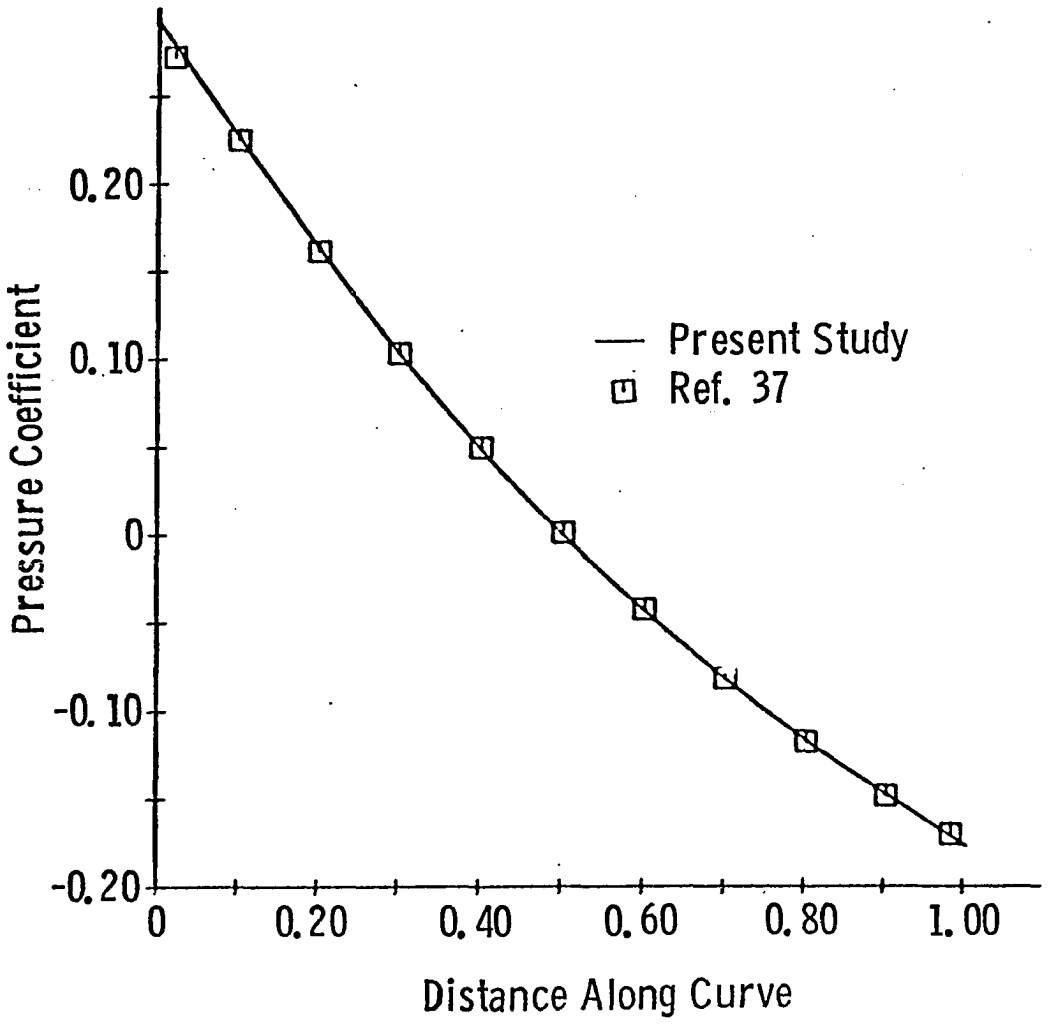


Figure 18. Single module ogive with geometric singularity on shock surface pressure coefficient; Mach number = 2, thickness = 0.1

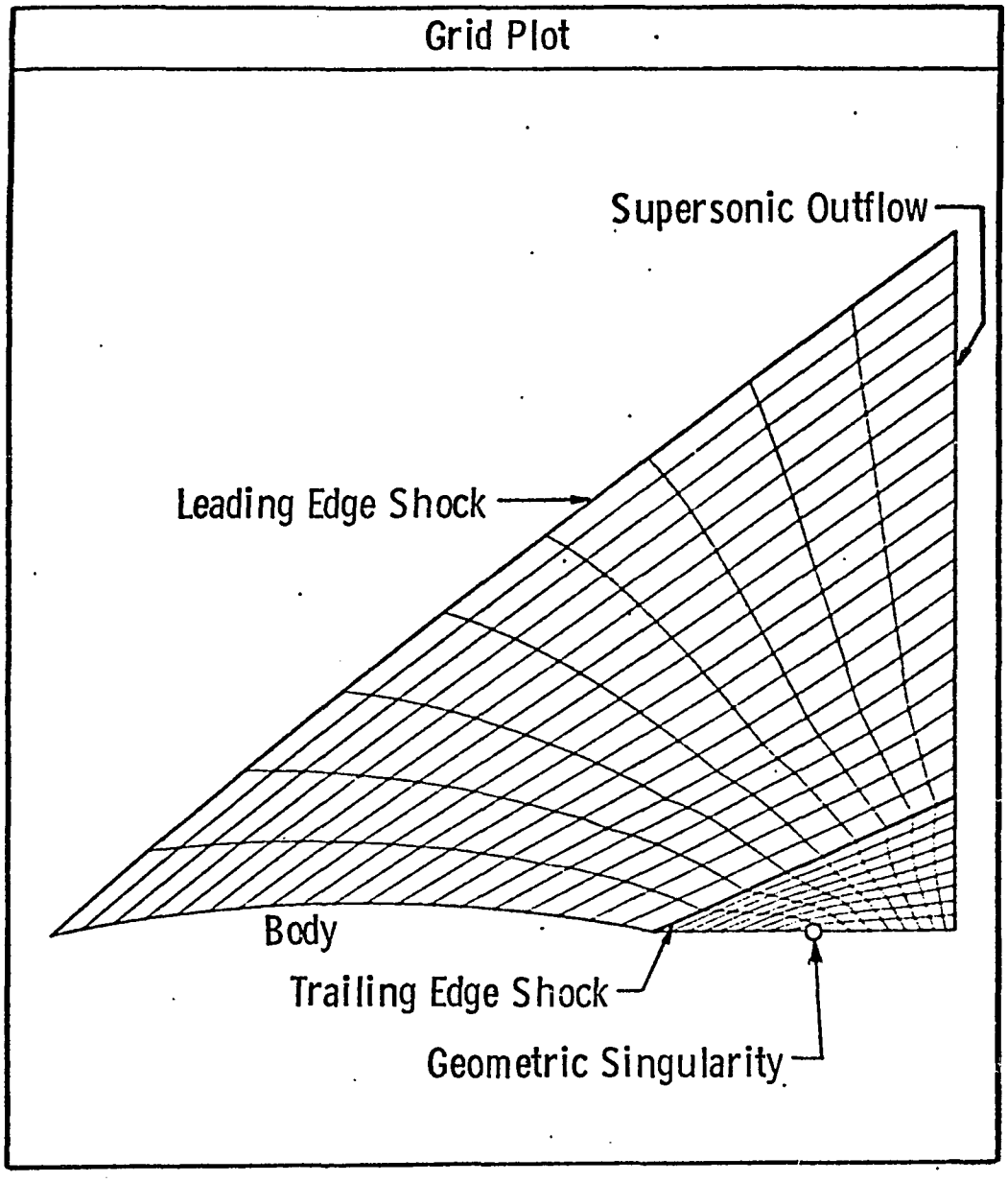


Figure 19. Double module ogive with aft region geometric singularity; Mach number = 2, thickness = 0.1

depicted in Figure 20 and compared to the shock capturing results of Schiff (37). Generally good agreement is observed where the comparison can be made. The present solution, however, exhibits a slightly higher estimate of the pressure just behind the trailing edge shock. This reflects the fact that the shock slope is slightly larger than it should be at this point. This same behavior is observed behind the leading edge shock and is due to some slight inconsistency in the scheme used to treat this type of corner point. The anomaly appearing near the center of the aft region symmetry boundary is due to the existence of the geometric singularity. This is contrary to the findings for the three-sided region in the single module ogive case where no anomaly appeared. A possible explanation of this follows. The grid for the single module case (Figure 17) appears to be near symmetric in some sense with respect to the singular point. Therefore, the truncation error in the solution, which is weakly dependent upon the problem geometry, will be essentially the same on either side of this point. Thus the solution will not be forced by the geometric contribution to the truncation error to exhibit a nonsmooth behavior through the singularity. On the other hand, this symmetry does not exist for the double module case as seen in Figure 19. This causes the truncation error on one side of the singular point to be different from that on the other side which gives rise to a weak perturbation on the solution at points near the singularity where the geometric variables are rapidly changing.

An additional test of the double module capability was conducted

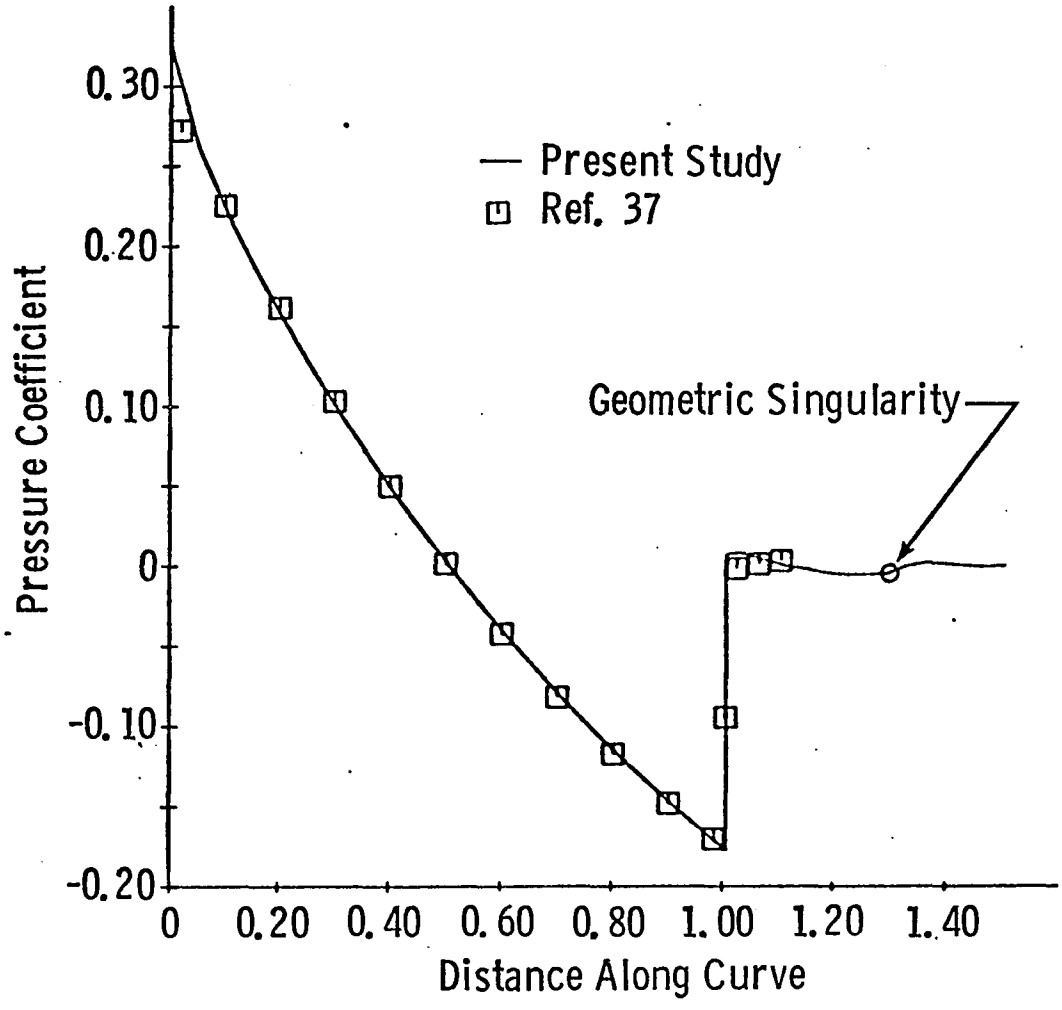


Figure 20. Double module ogive with aft region geometric singularity surface pressure coefficient; Mach number = 2, thickness = 0.1

on a single Mach reflection shock diffraction problem similar to the one shown in Figure 2. Region B is triangular and requires the use of a fictitious corner point (geometric singularity). This problem has a self-similar solution with respect to time as does the regular reflection problem of Figure 14. However, the existence of the slip surface boundary for this case results in a strange behavior of the grid as the solution unfolds. The grid points near the center of the slip surface tend to converge toward one another and drive the transformation Jacobian toward zero thus ultimately resulting in program failure. This is the only problem for which such behavior was observed. This behavior is especially peculiar in light of the manner in which the boundary grid point spacing is obtained. Boundaries which are nearly straight such as this slip surface should maintain approximately whatever relative grid point distributions they started out with. This is due to the use of Equation 33 of the previous chapter for the determination of the boundary point speed from the normal velocity component. Therefore, it is not clear what the mechanism is which forces these grid points together. The fact that the grid for this problem does not have a steady state solution but rather expands forever in time may play some role in the difficulty encountered but such a difficulty was not observed for the regular reflection problem which also has this type of unsteady grid. One major difference between the regular reflection problem which was easily computed and the single Mach reflection problem which could not be successfully computed is the existence of the slip surface interface boundary in the latter case. This might tend to suggest that the grid

problem is related to the procedure used for applying the slip surface boundary condition. However, it is the opinion of the author that the problem encountered is due to the failure of the boundary grid point spacing procedure to adequately treat this unsteady grid problem and that the successful solution of the regular reflection problem was fortuitous. Further research is necessary to prove or disprove this claim.

CHAPTER VIII. CONCLUDING REMARKS

A general method is presented for solving the unsteady two-dimensional Euler equations on multiple flow regions with arbitrarily-shaped and time-varying boundaries. The method is applicable to problems with moving boundaries provided the velocity of such movement may be determined or specified. This includes problems with moving pistons, structural deformations, accelerating bodies, moving or stationary discontinuity surfaces such as shocks and slip surfaces, etc. In the case of discontinuity surfaces, the scheme has the capability of capturing any discontinuities whose approximate shape and location is not known a priori provided the strength of such discontinuities is not excessive.

The resulting computer code may be used to solve a wide variety of two-dimensional flow problems. The multiple flow region capability may be used to compute one flow field with multiple regions or it may be used to simultaneously perform a parametric study of the solution for one flow problem.

The cause of the difficulty which was experienced with the grid for the single Mach reflection problem remains unknown, although it is believed to be related to the fact that this problem has no true steady-state solution. This problem will, therefore, require further research.

ACKNOWLEDGMENTS

The author thanks the NASA-Ames University Consortium and Engineering Research Institute of Iowa State University for partial support of this work.

Special thanks are given to the author's major Professor, Dr. Dale A. Anderson of Iowa State University, and to Dr. Paul Kutler of NASA-Ames Research Center, for their continued guidance and encouragement throughout the course of this research. The research and resulting career opportunities made available by Dr. Anderson and Dr. Kutler are very deeply appreciated.

Special thanks are also given to the author's wife, Julie, children, Michelle, Sophia, Jahmy, and Jessika, and parents, Gerald and Virginia whose continued support and constant faith have been invaluable contributions to this research effort.

REFERENCES

1. J. L. Steger. "Implicit finite-difference simulation of flow about arbitrary geometrics with application to airfoils." AIAA Paper No. 77-665, AIAA 10th Fluid and Plasmadynamics Conference, Albuquerque, New Mexico, June 27-29, 1977.

2. M. Inouye, J. V. Rakich and H. Lomax. "A description of numerical methods and computer programs for two-dimensional and axisymmetric supersonic flow over blunt-nosed and flared bodies." NASA TN D-2970, 1965.

3. F. B. Fuller. "Numerical solutions for supersonic flow of an ideal gas around blunt two-dimensional bodies." NASA TN D-791, 1961.

4. G. Ben-Dor and I. I. Glass. "Nonstationary oblique shock-wave reflections: actual isopycnics and numerical experiments." AIAA Journal, 16, No. 11 (November 1978), 1146-1153.

5. H. F. Ludloff and M. B. Friedman. "Aerodynamics of blasts-diffraction of blast around finite corners." Journal of the Aeronautical Sciences, 22, No. 1 (January 1955), 27-34.

6. J. F. Thompson, F. C. Thames and C. M. Mastin. "Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies." Journal of Computational Physics, 15 (1974), 299-319.

7. F. C. Thames, J. F. Thompson and C. M. Mastin. "Numerical solution of the Navier-Stokes equations for arbitrary two-dimensional airfoils." NASA SP-347, Part I, March 1975, 469-530.

8. U. Ghia and K. N. Ghia. "Numerical generation of a system of curvilinear coordinates for turbine cascade flow analysis." University of Cincinnati Report No. AFL 75-4-17, 1975.

9. A. A. Amsden and C. W. Hirt. "A simple scheme for generating curvilinear grids." Journal of Computational Physics, 11 (1973), 348-359.

10. V. P. Shapeev and Yu A. Shitov. "On a method of constructing curvilinear grids." Chisl. Met. Mekh. Spl. Sr., 7, No. 2 (1976).

11. W. D. Barfield. "Numerical method for generating orthogonal curvilinear meshes." Journal of Computational Physics, 5 (1970), 23-33.

12. W. H. Chu. "Development of a general finite difference approximation for a general domain." Journal of Computational Physics, 8 (1971), 392-408.

13. S. K. Godunov and G. P. Prokopov. "The use of moving meshes in gas-dynamical computations." U.S.S.R. Computational Mathematics and Mathematical Physics, 12 (1972), 182-195.

14. N. T. Danayev, V. D. Liseykin and N. N. Yanenko. "A method of nonstationary coordinates in gas dynamics." Foreign Technology Division, Wright-Patterson Air Force Base, Ohio, FTD-ID(RS)T-1673-78, October 1978.

15. V. M. Kovenya and N. N. Yanenko. "Implicit difference scheme on mobile grids for the numerical solution to Navier-Stokes equations of compressible gas." Foreign Technology Division, Wright-Patterson Air Force Base, Ohio, FTD-ID(RS)T-1671-78, October 1978.

16. N. N. Yanenko, E. Q. Kroshko, V. V. Liseikin, V. M. Fomin, V. P. Shapeev and Yu A. Shitov. "Methods for the construction of moving grids for problems of fluid dynamics with big deformations." Proceedings of the 5th International Conference on Numerical Methods in Fluid Dynamics, Enschede, Netherlands, June 28-July 3, 1976. Berlin, Germany: Springer-Verlag, 1976.

17. R. G. Hindman, Paul Kutler and Dale Anderson. "A two-dimensional unsteady Euler-equation solver for flow regions with arbitrary boundaries." AIAA Paper No. 79-1465, AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Virginia, July 23-24, 1979.

18. H. Viviand. "Conservative forms of gas dynamic equations." La Recherche Aerospatiale, No. 1 (January-February 1974), 65-68.

19. P. D. Lax. "Weak solutions for nonlinear hyperbolic equations and their numerical computation." Communications on Pure and Applied Mathematics, 7 (February 1954), 159-193.

20. P. D. Thomas and C. K. Lombard. "The geometric conservation law - a link between finite-difference and finite-volume methods of flow computation on moving grids." AIAA Paper No. 78-1208, AIAA 11th Fluid and Plasma Dynamics Conference, Seattle, Washington, July 10-12, 1978.

21. R. W. MacCormack. "The effect of viscosity in hypervelocity impact cratering." AIAA Paper No. 69-354, AIAA 7th Aerospace Sciences Meeting, Cincinnati, Ohio, April 30-May 2, 1969.

22. P. D. Thomas. "On the computation of boundary conditions in finite difference solutions for multi-dimensional inviscid flow fields." Lockheed Palo Alto Research Lab., Palo Alto, California, LMSC 6-82-71-3, March 1971.

23. C. K. Chu and A. Sereny. "Boundary conditions in finite difference fluid dynamic codes." Journal of Computational Physics, 15 (1974), 476-491.

24. P. D. Thomas, M. Vinokur, R. Bastianon and R. J. Conti. "Numerical solution for the three-dimensional inviscid supersonic flow of a blunt delta body." AIAA Journal, 10, No. 7 (July 1974), 887-894.

25. Ames Research Staff. "Equations, tables, and charts for compressible flow." NACA Report 1135, 1953.

26. P. Kutler. "Computation of three-dimensional, inviscid supersonic flows." Lecture Notes in Physics, Progress in Numerical Fluid Dynamics, No. 41. Berlin, Germany: Springer-Verlag, 1975.

27. V. Shankar, D. Anderson and P. Kutler. "Diffraction of a shock wave by a compression corner; Part II - single Mach reflection." AIAA Paper No. 77-89, AIAA 15th Aerospace Sciences Meeting, Los Angeles, California, January 24-26, 1977.

28. T. de Neef. "Treatment of boundaries in unsteady inviscid flow computations." Delft University of Technology Department of Aerospace Engineering Report LR-262, Netherlands, February 1978.

29. F. Marconi. "Internal corner flow fields." AIAA Paper No. 79-14, AIAA 17th Aerospace Sciences Meeting, New Orleans, Louisiana, January 15-17, 1979.

30. C. S. Kim. "Experimental studies of supersonic flow past a circular cylinder." Journal of the Physical Society of Japan, II (1956), 439-445.

31. F. S. Billig. "Shock-wave shapes around spherical and cylindrical-nosed bodies." Journal of Spacecraft and Rockets, 4, No. 6 (June 1967), 822-823.

32. J. E. Daywitt and D. A. Anderson. "Analysis of a time-dependent finite-difference technique for shock interaction and blunt-body flows." Iowa State University Engineering Research Institute Special Report No. 74074, May 1974.

33. A. W. Rizzi and M. Inouye. "A time-split finite-volume technique for three-dimensional blunt-body flow." AIAA Paper No. 73-133, AIAA 11th Aerospace Sciences Meeting, Washington, D. C., January 10-12, 1973.

34. G. Moretti. "Inviscid blunt body shock layers." Polytechnic Institute of Brooklyn PIBAL Report No. 68-15, June 1968.

35. Wayland Griffith. "Shock-tube studies of transonic flow over wedge profiles." Journal of the Aeronautical Sciences, 19, No. 4 (April 1952), 249-257.

36. P. Kutler and V. S. V. Shankar. "Diffraction of a shock wave by a compression corner; Part I - regular reflection." AIAA Paper No. 76-323, July 1976.

37. L. B. Schiff. Private communication, NASA-Ames Research Center, Moffett Field, California.

38. R. G. Hindman and D. A. Anderson. "Numerical solutions for inviscid supersonic flow about cones with large ellipticities." Iowa State University Engineering Research Institute Special Report No. 77334, September 1977.

39. M. Pandolfi. "Supersonic flow about elliptic cones with large semiaxis ratios." Instituto Di Macchine E Motori Per Aeromobili, Pubblicazione N. PP172, December 1975.

APPENDIX: ANALYSIS OF THE GRID PROPAGATION TECHNIQUE AND ITS COUPLING TO THE GOVERNING EQUATIONS

Consider the finite-difference representation of the strong conservation-law form of the Euler equations using MacCormack's scheme (21):

predictor:

$$(I\bar{q})^{\overline{n+1}} = (I\bar{q})^{n} - \Delta\tau\,\Delta_j\left[\{(y_\tau x_\eta^a - x_\tau y_\eta^a)\bar{q} + y_\eta^b\bar{f} - x_\eta^b\bar{g}\}^n\right]$$
$$- \Delta\tau\,\Delta_k\left[\{(x_\tau y_\xi^a - y_\tau x_\xi^a)\bar{q} - y_\xi^b\bar{f} + x_\xi^b\bar{g}\}^n\right] \qquad \text{A.1}$$

corrector:

$$(I\bar{q})^{n+1} = \frac{1}{2}\{(I\bar{q})^{\overline{n+1}} + (I\bar{q})^{n}$$
$$- \Delta\tau\,\nabla_j\left[\{(y_\tau x_\eta^a - x_\tau y_\eta^a)\bar{q} + y_\eta^b\bar{f} - x_\eta^b\bar{g}\}^{\overline{n+1}}\right]$$
$$- \Delta\tau\,\nabla_k\left[\{(x_\tau y_\xi^a - y_\tau x_\xi^a)\bar{q} - y_\xi^b\bar{f} + x_\xi^b\bar{g}\}^{\overline{n+1}}\right]\} \qquad \text{A.2}$$

where Δ_ℓ, ∇_ℓ are standard forward and backward finite-difference operators, respectively, on the ℓ-index {i.e., $\Delta_j[\phi] = (\phi_{j+1} - \phi_j)/\Delta\xi$, $\nabla_j[\phi] = (\phi_j - \phi_{j-1})/\Delta\xi$}. In order to gain precise repeatability of a uniform flow with such a scheme it is required that

$$\Delta_j\left[y_\eta^{b^n}\right] - \Delta_k\left[y_\xi^{b^n}\right] = \nabla_j\left[y_\eta^{b^{\overline{n+1}}}\right] - \nabla_k\left[y_\xi^{b^{\overline{n+1}}}\right] = 0 \qquad \text{A.3}$$

$$\Delta_j\left[x_\eta^{b^n}\right] - \Delta_k\left[x_\xi^{b^n}\right] = \nabla_j\left[x_\eta^{b^{\overline{n+1}}}\right] - \nabla_k\left[x_\xi^{b^{\overline{n+1}}}\right] = 0 \qquad \text{A.4}$$

and that I^{n+1} is obtained by solving the Jacobian identity equation with

the same finite-difference scheme used to difference the flow equations. That is,

predictor:

$$\overline{I^{n+1}} = I^n - \Delta\tau\, \Delta_j[(y_\tau x^a_\eta - x_\tau y^a_\eta)^n] - \Delta\tau\, \Delta_k[(x_\tau y^a_\xi - y_\tau x^a_\xi)^n] \qquad \text{A.5}$$

corrector:

$$I^{n+1} = \frac{1}{2}\{\overline{I^{n+1}} + I^n - \Delta\tau\, \nabla_j[(y_\tau x^a_\eta - x_\tau y^a_\eta)^{\overline{n+1}}] - \Delta\tau\, \nabla_k[(x_\tau y^a_\xi - y_\tau x^a_\xi)^{\overline{n+1}}]\} \qquad \text{A.6}$$

An inspection of Equations A.1-A.6 reveals the use of two of each of the metric quantities which are denoted by superscripts a and b. Although the a-metrics and the b-metrics represent the same physical quantities, they need not be numerically equivalent. This idea is exploited in the following derivation.

The conditions represented by Equations A.3 and A.4 always exist for a general geometry. They imply that the b-metrics should be evaluated so as to provide cancellation. This requires that

$$y^{b^n}_\eta = \Delta_k[y^n] \;,\; y^{b^{\overline{n+1}}}_\eta = \nabla_k[y^{\overline{n+1}}]$$

$$y^{b^n}_\xi = \Delta_j[y^n] \;,\; y^{b^{\overline{n+1}}}_\xi = \nabla_j[y^{\overline{n+1}}]$$

$$x_\eta^{b^n} = \Delta_k[x^n] \ , \ x_\eta^{\overline{b^{n+1}}} = \nabla_k[x^{\overline{n+1}}]$$

$$x_\xi^{b^n} = \Delta_j[x^n] \ , \ x_\xi^{\overline{b^{n+1}}} = \nabla_j[x^{\overline{n+1}}]$$

The determination of how the a-metrics should be computed is not such a simple matter. The condition represented by Equations A.5 and A.6 assumes the constraint that the quantities

$$(y_\tau x_\eta^a - x_\tau y_\eta^a)^n, \ (x_\tau y_\xi^a - y_\tau x_\xi^a)^n$$

in Equation A.5 and

$$(y_\tau x_\eta^a - x_\tau y_\eta^a)^{\overline{n+1}}, \ (x_\tau y_\xi^a - y_\tau x_\xi^a)^{\overline{n+1}}$$

in Equation A.6 are identical to their counterparts in Equations A.1 and A.2 respectively. However, nothing has been said about the evaluation of the individual quantities x_τ, y_τ, x_η^a, y_η^a, x_ξ^a, y_ξ^a. Clearly the determination of the a-metrics requires that the grid point coordinate field (x,y) be known. Suppose, then, that $(x,y)^{n+1}$ is determined by some scheme to be decided later. Suppose further that a test Jacobian, I_g, is defined as

$$I_g^{n+1} = \delta_j[x^{n+1}]\delta_k[y^{n+1}] - \delta_k[x^{n+1}]\delta_j[y^{n+1}] \qquad \text{A.7}$$

where δ_ℓ denotes the central difference operator on the ℓ-index. It is correct to require that the value of I^{n+1} computed from Equations A.5

and A.6 is in good agreement with the value of the test Jacobian. Thus it is required that

$$E = \|I_g^{n+1} - I^{n+1}\| \leq \varepsilon$$

where ε is a small number. The order of ε is controlled by two methods. First, the value of I_g^{n+1} is controlled by the scheme used to obtain $(x,y)^{n+1}$. Second, the value of I^{n+1} is dependent upon the manner in which the a-metrics are computed. Define $\hat{E} = I_g^{n+1} - I^{n+1}$. Then $E = \|\hat{E}\|$ and by substituting Equations A.5-A.7, $\hat{E}$ is written as

$$\begin{aligned}\hat{E} = {} & \delta_j[x^{n+1}]\delta_k[y^{n+1}] - \delta_k[x^{n+1}]\delta_j[y^{n+1}] - I^n \\ & + \frac{\Delta\tau}{2}\nabla_j[(y_\tau \bar{D}_k\, x - x_\tau \bar{D}_k\, y)^{\overline{n+1}}] \\ & + \frac{\Delta\tau}{2}\nabla_k[(x_\tau \bar{D}_j\, y - y_\tau \bar{D}_j\, x)^{\overline{n+1}}] \\ & + \frac{\Delta\tau}{2}\Delta_j[(y_\tau D_k\, x - x_\tau D_k\, y)^{n}] \\ & + \frac{\Delta\tau}{2}\Delta_k[(x_\tau D_j\, y - y_\tau D_j\, x)^{n}] \end{aligned} \qquad \text{A.8}$$

where the difference operators for the a-metrics, D_j, D_k, $\bar{D}_j$, $\bar{D}_k$, are to be determined. Since only the local error between I^{n+1} and I_g^{n+1} is of interest, it is correct to set $I^n = I_g^n$ and analyze the error obtained in only one step. In addition, suppose the coordinate field $(x,y)^{n+1}$ is obtained with the Euler predictor-modified Euler corrector scheme:

predictors:

$$\overline{x^{n+1}} = x^n + \Delta\tau\, x_\tau^n$$
$$\overline{y^{n+1}} = y^n + \Delta\tau\, y_\tau^n \qquad \text{A.9}$$

correctors:

$$x^{n+1} = \frac{1}{2}(x^n + \overline{x^{n+1}} + \Delta\tau\, \overline{x_\tau^{n+1}})$$
$$y^{n+1} = \frac{1}{2}(y^n + \overline{y^{n+1}} + \Delta\tau\, \overline{y_\tau^{n+1}})$$

then noting that $\delta_\ell\left[x^n + \frac{\Delta\tau}{2}x_\tau^n + \frac{\Delta\tau}{2}\overline{x_\tau^{n+1}}\right] = \delta_\ell[x^n] + \frac{\Delta\tau}{2}\delta_\ell[x_\tau^n] + \frac{\Delta\tau}{2}\delta_\ell[\overline{x_\tau^{n+1}}]$, Equation A.8 is rewritten as

$$\begin{aligned}
\hat{E} = {} & \{\delta_j[x^n] + \tfrac{\Delta\tau}{2}\delta_j[x_\tau^n] + \tfrac{\Delta\tau}{2}\delta_j[\overline{x_\tau^{n+1}}]\}\{\delta_k[y^n] \\
& + \tfrac{\Delta\tau}{2}\delta_k[y_\tau^n] + \tfrac{\Delta\tau}{2}\delta_k[\overline{y_\tau^{n+1}}]\} - \{\delta_k[x^n] + \tfrac{\Delta\tau}{2}\delta_k[x_\tau^n] \\
& + \tfrac{\Delta\tau}{2}\delta_k[\overline{x_\tau^{n+1}}]\}\{\delta_j[y^n] + \tfrac{\Delta\tau}{2}\delta_j[y_\tau^n] + \tfrac{\Delta\tau}{2}\delta_j[\overline{y_\tau^{n+1}}]\} \\
& - \delta_j[x^n]\delta_k[y^n] + \delta_k[x^n]\delta_j[y^n] \\
& + \tfrac{\Delta\tau}{2}\{\nabla_j[y_\tau^{n+1}\overline{D}_k[\overline{x^{n+1}}]] - \nabla_j[x_\tau^{n+1}\overline{D}_k[\overline{y^{n+1}}]] \\
& + \nabla_k[x_\tau^{n+1}\overline{D}_j[\overline{y^{n+1}}]] - \nabla_k[y_\tau^{n+1}\overline{D}_j[\overline{x^{n+1}}]] \\
& + \Delta_j[y_\tau^n D_k[x^n]] - \Delta_j[x_\tau^n D_k[y^n]] \\
& + \Delta_k[x_\tau^n D_j[y^n]] - \Delta_k[y_\tau^n D_j[x^n]]\}
\end{aligned}$$

which, after multiplying the quantities in braces and collecting terms, yields

$$\begin{aligned}
\hat{E} = \frac{\Delta\tau}{2}\{&\delta_j[x^n]\delta_k[y_\tau^n] + \delta_j[x^n]\delta_k[\overline{y_\tau^{n+1}}] \\
&+ \delta_j[x_\tau^n]\delta_k[y^n] + \delta_j[\overline{x_\tau^{n+1}}]\delta_k[y^n] \\
&- \delta_k[x^n]\delta_j[y_\tau^n] - \delta_k[x^n]\delta_j[\overline{y_\tau^{n+1}}] \\
&- \delta_k[x_\tau^n]\delta_j[y^n] - \delta_k[\overline{x_\tau^{n+1}}]\delta_j[y^n] \\
&+ \nabla_j[\overline{y_\tau^{n+1}}\overline{D}_k[\overline{x^{n+1}}]] - \nabla_j[\overline{x_\tau^{n+1}}\overline{D}_k[\overline{y^{n+1}}]] \\
&+ \nabla_k[\overline{x_\tau^{n+1}}\overline{D}_j[\overline{y^{n+1}}]] - \nabla_k[\overline{y_\tau^{n+1}}\overline{D}_j[\overline{x^{n+1}}]] \\
&+ \Delta_j[y_\tau^n D_k[x^n]] - \Delta_j[x_\tau^n D_k[y^n]] \\
&+ \Delta_k[x_\tau^n D_j[y^n]] - \Delta_k[y_\tau^n D_j[x^n]]\} \\
&+ \frac{\Delta\tau^2}{4}\{\delta_j[x_\tau^n]\delta_k[y_\tau^n] - \delta_j[y_\tau^n]\delta_k[x_\tau^n] \\
&+ \delta_j[x_\tau^n]\delta_k[\overline{y_\tau^{n+1}}] + \delta_j[\overline{x_\tau^{n+1}}]\delta_k[y_\tau^n] \\
&+ \delta_j[\overline{x_\tau^{n+1}}]\delta_k[\overline{y_\tau^{n+1}}] - \delta_k[x_\tau^n]\delta_j[\overline{y_\tau^{n+1}}] \\
&- \delta_k[\overline{x_\tau^{n+1}}]\delta_j[y_\tau^n] - \delta_k[\overline{x_\tau^{n+1}}]\delta_j[\overline{y_\tau^{n+1}}]\}
\end{aligned}$$

Examination of the first order term reveals the existence of the quantities $(x,y)^{\overline{n+1}}$ which are now replaced with the use of Equation A.9. The result is

$$\hat{E} = \frac{\Delta\tau}{2}\{\delta_j[x^n]\delta_k[y_\tau^n] + \delta_j[x^n]\delta_k[\overline{y_\tau^{n+1}}]$$

$$+ \delta_j[x_\tau^n]\delta_k[y^n] + \delta_j[\overline{x_\tau^{n+1}}]\delta_k[y^n]$$

$$- \delta_k[x^n]\delta_j[y_\tau^n] - \delta_k[x^n]\delta_j[\overline{y_\tau^{n+1}}]$$

$$- \delta_k[x_\tau^n]\delta_j[y^n] - \delta_k[\overline{x_\tau^{n+1}}]\delta_j[y^n]$$

$$+ \nabla_j[\overline{y_\tau^{n+1}}\overline{D}_k[x^n]] - \nabla_j[\overline{x_\tau^{n+1}}\overline{D}_k[y^n]]$$

$$+ \nabla_k[\overline{x_\tau^{n+1}}\overline{D}_j[y^n]] - \nabla_k[\overline{y_\tau^{n+1}}\overline{D}_j[x^n]]$$

$$+ \Delta_j[y_\tau^n D_k[x^n]] - \Delta_j[x_\tau^n D_k[y^n]]$$

$$+ \Delta_k[x_\tau^n D_j[y^n]] - \Delta_k[y_\tau^n D_j[x^n]]\}$$

$$+ \frac{\Delta\tau^2}{4}\{\delta_j[x_\tau^n]\delta_k[y_\tau^n] - \delta_j[y_\tau^n]\delta_k[x_\tau^n]$$

$$+ \delta_j[x_\tau^n]\delta_k[\overline{y_\tau^{n+1}}] + \delta_j[\overline{x_\tau^{n+1}}]\delta_k[y_\tau^n]$$

$$+ \delta_j[\overline{x_\tau^{n+1}}]\delta_k[\overline{y_\tau^{n+1}}] - \delta_k[x_\tau^n]\delta_j[\overline{y_\tau^{n+1}}]$$

$$- \delta_k[\overline{x_\tau^{n+1}}]\delta_j[y_\tau^n] - \delta_k[\overline{x_\tau^{n+1}}]\delta_j[\overline{y_\tau^{n+1}}]$$

$$+ 2(\nabla_j[\overline{y_\tau^{n+1}}\overline{D}_k[x_\tau^n]] - \nabla_j[\overline{x_\tau^{n+1}}\overline{D}_k[y_\tau^n]]$$

$$+ \nabla_k[\overline{x_\tau^{n+1}}\overline{D}_j[y_\tau^n]] - \nabla_k[\overline{y_\tau^{n+1}}\overline{D}_j[x_\tau^n]])\} \qquad \text{A.10}$$

However, the terms involving the undetermined operators, D_j, D_k, $\overline{D}_j$, $\overline{D}_k$, are of the form $\nabla_\ell[(AB)_\ell]$ or $\Delta_\ell[(AB)_\ell]$ which, upon expansion, become

$$\nabla_{\ell}[A_{\ell}B_{\ell}] = (\nabla_{\ell}[A_{\ell}])B_{\ell-1} + A_{\ell}(\nabla_{\ell}[B_{\ell}])$$

$$\Delta_{\ell}[A_{\ell}B_{\ell}] = (\Delta_{\ell}[A_{\ell}])B_{\ell+1} + A_{\ell}(\Delta_{\ell}[B_{\ell}])$$

or

$$\nabla_{\ell}[A_{\ell}B_{\ell}] = (\nabla_{\ell}[A_{\ell}])(S_{\ell}^{-}[B_{\ell}]) + A_{\ell}(\nabla_{\ell}[B_{\ell}])$$

$$\Delta_{\ell}[A_{\ell}B_{\ell}] = (\Delta_{\ell}[A_{\ell}])(S_{\ell}^{+}[B_{\ell}]) + A_{\ell}(\Delta_{\ell}[B_{\ell}]) \qquad \text{A.11}$$

where $S_{\ell}^{\pm}$ is the shift operator on the ℓ-index. Also recognizing that the terms $x_{\tau}^{\overline{n+1}}$, $y_{\tau}^{\overline{n+1}}$ may be replaced by their Taylor expansions,

$$x_{\tau}^{\overline{n+1}} = x_{\tau}^{n} + \Delta\tau\, x_{\tau\tau}^{n} + O(\Delta\tau^2)$$

$$y_{\tau}^{\overline{n+1}} = y_{\tau}^{n} + \Delta\tau\, y_{\tau\tau}^{n} + O(\Delta\tau^2)$$

Equation A.10 becomes

$$\begin{aligned} E = \frac{\Delta\tau}{2}\{&2(\delta_j[x^n]\delta_k[y_{\tau}^{n}] - \delta_j[y_{\tau}^{n}]\delta_k[x^n] \\ &+ \delta_j[x_{\tau}^{n}]\delta_k[y^n] - \delta_k[x_{\tau}^{n}]\delta_j[y^n]) \\ &- \nabla_k[y_{\tau}^{n}]S_k^{-}[\overline{D}_j[x^n]] - \Delta_k[y_{\tau}^{n}]S_k^{+}[D_j[x^n]] \\ &- \nabla_j[x_{\tau}^{n}]S_j^{-}[\overline{D}_k[y^n]] - \Delta_j[x_{\tau}^{n}]S_j^{+}[D_k[y^n]] \\ &+ \nabla_j[y_{\tau}^{n}]S_j^{-}[\overline{D}_k[x^n]] + \Delta_j[y_{\tau}^{n}]S_j^{+}[D_k[x^n]] \\ &+ \nabla_k[x_{\tau}^{n}]S_k^{-}[\overline{D}_j[y^n]] + \Delta_k[x_{\tau}^{n}]S_k^{+}[D_j[y^n]] \end{aligned}$$

$$+ y_\tau^n(\nabla_j[\bar{D}_k[x^n]] + \Delta_j[D_k[x^n]]$$

$$- \nabla_k[\bar{D}_j[x^n]] - \Delta_k[D_j[x^n]])$$

$$+ x_\tau^n(\nabla_k[\bar{D}_j[y^n]] + \Delta_k[D_j[y^n]]$$

$$- \nabla_j[\bar{D}_k[y^n]] - \Delta_j[D_k[y^n]])\}$$

$$+ \frac{\Delta\tau^2}{2}\{2(\delta_j[x_\tau^n]\delta_k[y_\tau^n] - \delta_j[y_\tau^n]\delta_k[x_\tau^n])$$

$$+ \nabla_j[y_\tau^n\bar{D}_k[x_\tau^n]] + \nabla_k[x_\tau^n\bar{D}_j[y_\tau^n]]$$

$$- \nabla_j[x_\tau^n\bar{D}_k[y_\tau^n]] - \nabla_k[y_\tau^n\bar{D}_j[x_\tau^n]]$$

$$+ \delta_k[y^n]\delta_j[x_{\tau\tau}^n] - \nabla_j[x_{\tau\tau}^n\bar{D}_k[y^n]]$$

$$+ \delta_j[x^n]\delta_k[y_{\tau\tau}^n] + \nabla_k[x_{\tau\tau}^n\bar{D}_j[y^n]]$$

$$- \delta_k[x^n]\delta_j[y_{\tau\tau}^n] + \nabla_j[y_{\tau\tau}^n\bar{D}_k[x^n]]$$

$$- \delta_j[y^n]\delta_k[x_{\tau\tau}^n] - \nabla_k[y_{\tau\tau}^n\bar{D}_j[x^n]]\}$$

$$+ O(\Delta\tau^3) \qquad \text{A.12}$$

At this point the operators, D_j, D_k, $\bar{D}_j$, $\bar{D}_k$, are chosen to make the first order term identically zero. This is accomplished by setting

$$D_j = S_k^-\delta_j$$

$$D_k = S_j^-\delta_k$$

$$\bar{D}_j = S_k^+\delta_j$$

$$\bar{D}_k = S_j^+ \delta_k \qquad \text{A.13}$$

For example, the first line of Equation A.12 becomes

$$2\delta_j[x^n]\delta_k[y_\tau^n] - \nabla_k[y_\tau^n]S_k^- S_k^+ \delta_j[x^n] - \Delta_k[y_\tau^n]S_k^+ S_k^- \delta_j[x^n]$$

which simplifies to

$$(2\delta_k - \nabla_k - \Delta_k)\,[y_\tau^n]\delta_j[x^n]$$

since $S_\ell^+ S_\ell^- \equiv 1$. But the operator $(2\delta_k - \nabla_k - \Delta_k)$ is the null operator. That is,

$$(2\delta_k - \nabla_k - \Delta_k)[\phi] = (\phi_{k+1} - \phi_{k-1}) - (\phi_k - \phi_{k-1}) - (\phi_{k+1} - \phi_k)$$

$$= 0 \quad \text{for any function, } \phi.$$

This result holds for the first four lines of Equation A.12. The last two lines of the first order term are similar to each other in form. For example, the quantity inside parentheses in the last line of this term is written as

$$\nabla_k[S_k^+\delta_j[y^n]\,] + \Delta_k[S_k^-\delta_j[y^n]\,] - \nabla_j[S_j^+\delta_k[y^n]\,] - \Delta_j[S_j^-\delta_k[y^n]\,]$$

upon substitution of Equation A.13. Since $\nabla_k S_k^+ = \Delta_k$ and $\Delta_k S_k^- = \nabla_k$, the two positive terms in this expression are simplified to

$$\Delta_k[\delta_j[y^n]\,] + \nabla_k[\delta_j[y^n]\,] = (\Delta_k + \nabla_k)[\delta_j[y^n]\,]$$

which becomes simply $2\delta_k[\delta_j[y^n]]$. Similarly, the negative terms simplify to $-2\delta_j[\delta_k[y^n]]$. Since the operators δ_j, δ_k commute, these two terms cancel identically. In a similar fashion the remaining line of the first order term of Equation A.12 can be shown to be identically zero. Thus, $\hat{E}$ is at most $O(\Delta\tau^2)$ provided the proper choice is made for the operators, D_j, $\bar{D}_j$, D_k, $\bar{D}_k$. Substitution of Equation A.13 into the remainder of Equation A.12 yields

$$\begin{aligned}\hat{E} = \frac{\Delta\tau^2}{2}\{&2(\delta_j[x_\tau^n]\delta_k[y_\tau^n] - \delta_j[y_\tau^n]\delta_k[x_\tau^n]) \\ &+ \nabla_j[y_\tau^n S_j^+ \delta_k[x_\tau^n]] + \nabla_k[x_\tau^n S_k^+ \delta_j[y_\tau^n]] \\ &- \nabla_j[x_\tau^n S_j^+ \delta_k[y_\tau^n]] - \nabla_k[y_\tau^n S_k^+ \delta_j[x_\tau^n]] \\ &+ \delta_k[y^n]\delta_j[x_{\tau\tau}^n] - \nabla_j[x_{\tau\tau}^n S_j^+ \delta_k[y^n]] \\ &+ \delta_j[x^n]\delta_k[y_{\tau\tau}^n] + \nabla_k[x_{\tau\tau}^n S_k^+ \delta_j[y^n]] \\ &- \delta_k[x^n]\delta_j[y_{\tau\tau}^n] + \nabla_j[y_{\tau\tau}^n S_j^+ \delta_k[x^n]] \\ &- \delta_j[y^n]\delta_k[x_{\tau\tau}^n] - \nabla_k[y_{\tau\tau}^n S_k^+ \delta_j[x^n]]\} \\ &+ O(\Delta\tau^3)\end{aligned} \qquad \text{A.14}$$

With the use of Equation A.11, Equation A.14 is rewritten as

$$\hat{E} = \frac{\Delta\tau^2}{2}\{2(\delta_j[x_\tau^n]\delta_k[y_\tau^n] - \delta_j[y_\tau^n]\delta_k[x_\tau^n])$$

$$+ \nabla_j[y_\tau^n]\delta_k[x_\tau^n] + y_\tau^n\Delta_j[\delta_k[x_\tau^n]]$$

$$+ \nabla_k[x_\tau^n]\delta_j[y_\tau^n] + x_\tau^n\Delta_k[\delta_j[y_\tau^n]]$$

$$- \nabla_j[x_\tau^n]\delta_k[y_\tau^n] - x_\tau^n\Delta_j[\delta_k[y_\tau^n]]$$

$$- \nabla_k[y_\tau^n]\delta_j[x_\tau^n] - y_\tau^n\Delta_k[\delta_j[x_\tau^n]]$$

$$+ \delta_k[y^n]\delta_j[x_{\tau\tau}^n] - \nabla_j[x_{\tau\tau}^n]\delta_k[y^n]$$

$$- x_{\tau\tau}^n\Delta_j[\delta_k[y^n]] + x_{\tau\tau}^n\Delta_k[\delta_j[y^n]]$$

$$+ \delta_j[x^n]\delta_k[y_{\tau\tau}^n] + \nabla_k[x_{\tau\tau}^n]\delta_j[y^n]$$

$$- \delta_k[x^n]\delta_j[y_{\tau\tau}^n] + \nabla_j[y_{\tau\tau}^n]\delta_k[x^n]$$

$$+ y_{\tau\tau}^n\Delta_j[\delta_k[x^n]] - y_{\tau\tau}^n\Delta_k[\delta_j[x^n]]$$

$$- \delta_j[y^n]\delta_k[x_{\tau\tau}^n] - \nabla_k[y_{\tau\tau}^n]\delta_j[x^n]\}$$

$$+ O(\Delta\tau^3) \qquad \text{A.15}$$

Now with the introduction of two new operators, Q_ℓ, T;

$$Q_\ell = \delta_\ell - \nabla_\ell$$

$$T = \Delta_k\delta_j - \Delta_j\delta_k$$

and the collection and reorganization of terms, Equation A.15 becomes

$$\hat{E} = \frac{\Delta\tau^2}{2}\{x_\tau^n T[y_\tau^n] - y_\tau^n T[x_\tau^n] + x_{\tau\tau}^n T[y^n] - y_{\tau\tau}^n T[x^n]$$

$$+ \frac{1}{2}(\Delta_j[x_\tau^n]\Delta_k[y_\tau^n] - \nabla_j[x_\tau^n]\nabla_k[y_\tau^n])$$

$$- \frac{1}{2}(\Delta_j[y_\tau^n]\Delta_k[x_\tau^n] - \nabla_j[y_\tau^n]\nabla_k[x_\tau^n])$$

$$+ \delta_k[y^n]Q_j[x_{\tau\tau}^n] - \delta_j[y^n]Q_k[x_{\tau\tau}^n]$$

$$+ \delta_j[x^n]Q_k[y_{\tau\tau}^n] - \delta_k[x^n]Q_j[y_{\tau\tau}^n]\}$$

$$+ O(\Delta\tau^3)$$

It is a simple matter to show that $T \simeq O(\Delta\xi,\Delta\eta)$, $Q_j \simeq O(\Delta\xi)$, and $Q_k \simeq O(\Delta\eta)$. In addition the quantities with the form $\Delta_j[\phi]\Delta_k[\psi] - \nabla_j[\phi]\nabla_k[\psi]$ are easily shown to be $O(\Delta\xi,\Delta\eta)$. As a result, the local error, $\hat{E}$, between I^{n+1} and I_g^{n+1} reduces to

$$\hat{E} = O(\Delta\tau^3,\Delta\xi\Delta\tau^2,\Delta\eta\Delta\tau^2)$$

This completes the derivation and at this point a summary of the two major results is given. First, the coordinate field $(x,y)^{n+1}$ may be obtained with the Euler predictor-modified Euler corrector scheme repeated here.

predictors:

$$\overline{x^{n+1}} = x^n + \Delta\tau\, x_\tau^n$$

$$\overline{y^{n+1}} = y^n + \Delta\tau\, y_\tau^n$$

correctors:

$$x^{n+1} = \frac{1}{2}(x^n + x^{\overline{n+1}} + \Delta\tau\, x_\tau^{\overline{n+1}})$$

$$y^{n+1} = \frac{1}{2}(y^n + y^{\overline{n+1}} + \Delta\tau\, y_\tau^{\overline{n+1}}) \qquad \text{A.16}$$

Second, the a-metrics appearing in Equations A.1-A.2 and A.5-A.6 are computed by the formulae;

$$y_\eta^{a^n} = S_j^- \delta_k [y^n] \;,\; y_\eta^{a^{\overline{n+1}}} = S_j^+ \delta_k [y^{\overline{n+1}}]$$

$$y_\xi^{a^n} = S_k^- \delta_j [y^n] \;,\; y_\xi^{a^{\overline{n+1}}} = S_k^+ \delta_j [y^{\overline{n+1}}]$$

$$x_\eta^{a^n} = S_j^- \delta_k [x^n] \;,\; x_\eta^{a^{\overline{n+1}}} = S_j^+ \delta_k [x^{\overline{n+1}}]$$

$$x_\xi^{a^n} = S_k^- \delta_j [x^n] \;,\; x_\xi^{a^{\overline{n+1}}} = S_k^+ \delta_j [x^{\overline{n+1}}]$$

The employment of these two results provides assurance that any grid resulting from the application of Equation A.16 will be properly represented by the integrated Jacobian value obtained from Equations A.5-A.6, thus a necessary consistency is achieved.

Thus far, the problem of determining the grid point speeds, x_τ^n, y_τ^n, $x_\tau^{\overline{n+1}}$, $y_\tau^{\overline{n+1}}$, for use in Equations A.1-A.2, A.5-A.6, and A.16 has been avoided. This problem is now addressed.

Suppose the coordinate field $(x,y)^n$ satisfies the equations

$$G^n[x] = 0 \ , \ G^n[y] = 0$$

where G^n is just some operator (nonlinear in general) used for determining $(x,y)^n$. Further suppose that it is desirable to have $(x,y)^{n+1}$ satisfying

$$G^{n+1}[x] = \varepsilon_1 \ , \ G^{n+1}[y] = \varepsilon_2$$

where ε_1, ε_2 are as close to zero as possible. Then provided that G is of such a nature that $(x_\tau,y_\tau)^n$ and $(x_\tau,y_\tau)^{\overline{n+1}}$ may be determined by requiring $G^n_\tau[x] = 0$, $G^n_\tau[y] = 0$ and $G^{\overline{n+1}}_\tau[x] = 0$, $G^{\overline{n+1}}_\tau[y] = 0$ respectively, the operator G^{n+1} may be expressed as

$$G^{n+1} = G^n + \frac{\Delta\tau}{2}G^n_\tau + \frac{\Delta\tau}{2}G^{\overline{n+1}}_\tau + O(\Delta\tau^3)$$

Now, since $(x,y)^n$ is that coordinate field satisfying $G^n[x] = 0$, $G^n[y] = 0$, and $(x_\tau,y_\tau)^n$, $(x_\tau,y_\tau)^{\overline{n+1}}$ are obtained by requiring $G^n_\tau[x] = 0$, $G^n_\tau[y] = 0$, and $G^{\overline{n+1}}_\tau[x] = 0$, $G^{\overline{n+1}}_\tau[y] = 0$ respectively, then $(x,y)^{n+1}$ must satisfy

$$G^{n+1}[x] = O(\Delta\tau^3) \ , \ G^{n+1}[y] = O(\Delta\tau^3)$$

Thus ε_1, $\varepsilon_2 \sim O(\Delta\tau^3)$ which is quite satisfactory. If the operator, G, is taken to be the one introduced by Thompson, Thames, and Mastin (6) then

$$G = \alpha\frac{\partial^2}{\partial\xi^2} - 2\beta\frac{\partial^2}{\partial\xi\partial\eta} + \gamma\frac{\partial^2}{\partial\eta^2}$$

$$+ I^2\{P(\tau,\xi,\eta)\frac{\partial}{\partial\xi} + Q(\tau,\xi,\eta)\frac{\partial}{\partial\eta}\}$$

where

$$\alpha = x_\eta^2 + y_\eta^2$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^2 + y_\xi^2$$

and

$$G_\tau = \alpha_\tau \frac{\partial^2}{\partial \xi^2} - 2\beta_\tau \frac{\partial^2}{\partial \xi \partial \eta} + \gamma_\tau \frac{\partial^2}{\partial \eta^2}$$

$$+ (I^2 P_\tau + 2IPI_\tau)\frac{\partial}{\partial \xi} + (I^2 Q_\tau + 2IQI_\tau)\frac{\partial}{\partial \eta}$$

$$+ \alpha \frac{\partial^2 \frac{\partial}{\partial \tau}}{\partial \xi^2} - 2\beta \frac{\partial^2 \frac{\partial}{\partial \tau}}{\partial \xi \partial \eta} + \gamma \frac{\partial^2 \frac{\partial}{\partial \tau}}{\partial \eta^2}$$

$$+ I^2 \{P(\tau,\xi,\eta)\frac{\partial \frac{\partial}{\partial \tau}}{\partial \xi} + Q(\tau,\xi,\eta)\frac{\partial \frac{\partial}{\partial \tau}}{\partial \eta}\}$$

Substitution of

$$\alpha_\tau = 2x_\eta (x_\tau)_\eta + 2y_\eta (y_\tau)_\eta$$

$$\beta_\tau = x_\xi (x_\tau)_\eta + (x_\tau)_\xi x_\eta + y_\xi (y_\tau)_\eta + (y_\tau)_\xi y_\eta$$

$$\gamma_\tau = 2x_\xi (x_\tau)_\xi + 2y_\xi (y_\tau)_\xi$$

$$I_\tau = x_\xi (y_\tau)_\eta + (x_\tau)_\xi y_\eta - (x_\tau)_\eta y_\xi - x_\eta (y_\tau)_\xi$$

and collecting terms yields

$$
\begin{aligned}
G_\tau[\cdot] = {} & G\left[\frac{\partial}{\partial\tau}[\cdot]\right] \\
& + \{2x_\eta \frac{\partial^2}{\partial\xi^2}[\cdot] - 2x_\xi \frac{\partial^2}{\partial\xi\partial\eta}[\cdot] \\
& - 2Iy_\xi(P\frac{\partial}{\partial\xi}[\cdot] + Q\frac{\partial}{\partial\eta}[\cdot])\}(x_\tau)_\eta \\
& + \{2y_\eta \frac{\partial^2}{\partial\xi^2}[\cdot] - 2y_\xi \frac{\partial^2}{\partial\xi\partial\eta}[\cdot] \\
& + 2Ix_\xi(P\frac{\partial}{\partial\xi}[\cdot] + Q\frac{\partial}{\partial\eta}[\cdot])\}(y_\tau)_\eta \\
& + \{2x_\xi \frac{\partial^2}{\partial\eta^2}[\cdot] - 2x_\eta \frac{\partial^2}{\partial\xi\partial\eta}[\cdot] \\
& + 2Iy_\eta(P\frac{\partial}{\partial\xi}[\cdot] + Q\frac{\partial}{\partial\eta}[\cdot])\}(x_\tau)_\xi \\
& + \{2y_\xi \frac{\partial^2}{\partial\eta^2}[\cdot] - 2y_\eta \frac{\partial^2}{\partial\xi\partial\eta}[\cdot] \\
& - 2Ix_\eta(P\frac{\partial}{\partial\xi}[\cdot] + Q\frac{\partial}{\partial\eta}[\cdot])\}(y_\tau)_\xi \\
& + I^2(P_\tau \frac{\partial}{\partial\xi}[\cdot] + Q_\tau \frac{\partial}{\partial\eta}[\cdot])
\end{aligned}
\qquad \text{A.17}
$$

Now suppose $(x,y)^n$ are known. Then application of Equation A.17 to both x and y yields two linear coupled partial differential equations in the dependent variables $(x_\tau,y_\tau)^n$. The variable coefficients are computable as various combinations of the metrics and their spatial variations. Application of a finite-difference approximation to these equations results in a set of linear algebraic equations to which a direct solution may be obtained. Equation A.17 is also applicable to

$(x,y)^{\overline{n+1}}$ which yields the speeds $(x_\tau,y_\tau)^{\overline{n+1}}$. The correct boundary condition for Equation A.17 is that (x_τ,y_τ) be specified at all boundary points. This is a natural boundary condition since boundaries are typically walls, bodies, shocks, etc., whose speed is either known or may be determined.